

UNIVERSIDADE FEDERAL DO PARANÁ

EDEMIR REGINALDO MACIEL

APRISIONAMENTO NA INFRAESTRUTURA DE  
TECNOLOGIA DA INFORMAÇÃO

CURITIBA PR

2018

EDEMIR REGINALDO MACIEL

APRISIONAMENTO NA INFRAESTRUTURA DE  
TECNOLOGIA DA INFORMAÇÃO

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática, no Programa de Pós-Graduação em Informática, setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Dra. Leticia Mara Peres.

CURITIBA PR

2018

Catálogo na Fonte: Sistema de Bibliotecas, UFPR  
Biblioteca de Ciência e Tecnologia

M152a

Maciel, Edemir Reginaldo

Aprisionamento na infraestrutura de tecnologia da informação / Edemir Reginaldo Maciel. – Curitiba, 2018.

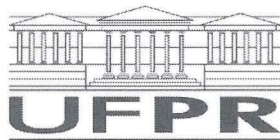
Dissertação - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática, 2018.

Orientadora: Leticia Mara Peres. -

1.Tecnologia da informação. 2. Governança corporativa. 3.  
Aprisionamento tecnológico. I. Universidade Federal do Paraná. II. Peres,  
Leticia Mara. III. Título.

CDD: 658.4038

Bibliotecária: Vanusa Maciel - CRB - 9/1928



MINISTÉRIO DA EDUCAÇÃO  
SETOR CIÊNCIAS EXATAS  
UNIVERSIDADE FEDERAL DO PARANÁ  
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO INFORMÁTICA

## TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **EDEMIR REGINALDO MACIEL** intitulada: **APRISIONAMENTO NA INFRAESTRUTURA DE TECNOLOGIA DA INFORMAÇÃO**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 01 de Março de 2018.



LETICIA MARA PERES

Presidente da Banca Examinadora (UFPR)



MARCOS SFAIR SUNYE

Avaliador Interno (UFPR)



MARIA SALETE MARCON GOMES VAZ

Avaliador Externo (UEPG)



*Dedico este trabalho para Ferceia,  
que é o amor da minha vida!*

# Agradecimentos

Agradeço muito a minha orientadora, professora Dra. Letícia Mara Peres, pela compreensão, dedicação e por tornar este trabalho possível.

Agradeço ao Centro de Computação Científica e Software Livre (C3SL) do Departamento de Informática da UFPR pela colaboração em fornecer dados e informações para os estudos de caso. Em especial, e em nome da professora Dra. Letícia Mara Peres e do professor Dr. Marcos Sfair Sunye, agradeço a todos os professores que integram o C3SL pelas constantes colaborações, pelos apoios e pela inspiração que me faz acreditar na dimensão civilizatória que existe em cada um deles - por exercerem com tanta dedicação a atividade de docência.

Aos colegas alunos que contribuíram com ideias e estiveram juntos comigo ao longo desta caminhada, ensinando-me a importância de ter amigos. Aos colegas de trabalho que compartilharam os desafios que surgiram durante o tempo em que eu me dedicava aos estudos.

Agradeço de coração a Dona Maria Lúcia e ao Seu Osmar José, tão queridos e tão presentes em todos os momentos. Aos meus pais Rita de Fátima e José Irineu que, para além do dom da vida, me ensinaram a persistir - sempre com humildade, coragem e determinação.

Por fim, agradeço a Ferceia, minha amada alma gêmea, pelo apoio incondicional e as minhas filhas Maria Clara e Helena de Fátima, que me fazem sonhar e lutar por um futuro melhor.

# Resumo

A capacidade da governança corporativa, de decidir e implementar mudanças de forma ágil, eficiente e eficaz, é o que torna possível a existência de organizações que atendam suas missões. Se a tomada de decisão está submetida a elevado grau de aprisionamento, essa organização tem limitada a capacidade de executar as alterações necessárias à sobrevivência. Esse aprisionamento também ocorre na governança e gestão da Tecnologia da Informação (TI). Os ativos da infraestrutura de TI determinam o grau de mobilidade que uma organização terá, ou seja, o grau de aprisionamento no qual ela se encontra ou se encontrará. O presente trabalho analisa, a partir da governança e gestão de TI, os principais ativos da infraestrutura de TI que elevam ou diminuem o grau de aprisionamento - e propõe um Índice para análise desse aprisionamento. Baseado na governança corporativa, e ancorado em elementos endógenos de governança e de gestão de TI, pretende-se com esse Índice observar o fenômeno do aprisionamento - especificamente nos ativos da sua infraestrutura. Por fim, nos estudos de caso realizados, foi possível calcular o Índice de aprisionamento da organização envolvida e os principais elementos que causam esse fenômeno.

**Palavras-chave:** Infraestrutura de Tecnologia da Informação; Gestão de Tecnologia da Informação; Aprisionamento Tecnológico; Lock-in.

# Abstract

The ability of corporate governance to decide and implement changes in an agile, efficient and effective way is what makes possible the existence of organizations that fulfill their missions. A decision-making that rises with a high degree of lock-in, limiteds the organization in its ability to perform the changes necessary for its survival. This lock-in also occurs on the context of governance and management of Information Technology (TI). IT infrastructure assets determine the degree of mobility an organization will have, that is, the degree of lock-in it is in or will encounter. The present work analyzes, from the IT governance and management, the main assets of the IT infrastructure that increase or decrease the degree of lock-in and, it proposes an index for analysis of this lock-in. Based on corporate governance, anchored in endogenous elements of IT governance and management, it is intended with this index to observe the phenomenon of lock-in, specifically in the assets of its infrastructure. Finally, in the case studies that are carried out, it was possible to demonstrate the degree of lock-in of organization and the main elements that cause this phenomenon.

**Keywords:** Information Technology Infrastructure; Information Technology Management; Lock-in.



# Lista de Figuras

2.1	Modelo de governança empresarial. Fonte: [1] . . . . .	16
2.2	Modelo de alinhamento estratégico. Fonte: [2] . . . . .	20
2.3	Decisões fundamentais em TI. Fonte: [3] . . . . .	21
2.4	Características e preocupações do gerenciamento operacional e estratégico. Fonte: [4] . . . . .	22
2.5	Componentes da infraestrutura de TI. Fonte: [5] . . . . .	24
2.6	Os componentes de um sistema de informação. Fonte: [6] . . . . .	26
2.7	CPU e memória principal conectadas via barramento. Fonte: [7] . . . . .	27
2.8	Esquema do cluster com acesso externo. Fonte:[8]. . . . .	29
2.9	Uma visão geral do software. Fonte:[6] . . . . .	31
2.10	Tendências em software. Fonte:[6] . . . . .	33
2.11	Ciclo de aprisionamento. Fonte: [9] . . . . .	46
2.12	Graus dos custos de troca e aprisionamento Fonte:[10] . . . . .	48
2.13	Tipo de aprisionamento e custos de troca associados. Fonte: [9] . . . . .	49
4.1	Descrição do hardware e do software segundo o edital. Fonte:[11] . . . . .	74

# Lista de Tabelas

2.1	Categoria de licenças . . . . .	35
2.2	Tipos de licenças na categoria das licenças permissivas . . . . .	36
2.3	Tipos de licenças na categoria das licenças recíprocas totais. . . . .	37
2.4	Tipos de licenças na categoria das licenças recíprocas parciais . . . . .	38
2.5	Tipos de ferramentas para analisar as licenças existentes em um software. . . . .	39
2.6	Exemplos de método, modelo e/ou norma. Fonte: o autor com base na ISO/IEC 9126, 14598, 25000, Paradigma GQM, Modelo Rocha e Método Medeiros Pros . . . . .	41
2.7	Características e subcaracterísticas da qualidade de software.Fonte: [12] . . . . .	44
3.1	Elementos dos recursos de hardware considerados críticos ao aprisionamento.Fonte: o autor . . . . .	55
3.2	Características da qualidade de software selecionadas para apoiar a identificação do aprisionamento no recurso de software. Fonte:o autor com base na ISO/IEC 25010	59
3.3	Elementos dos recursos de hardware considerados críticos ao aprisionamento . .	62
3.4	Elementos dos recursos de software considerados críticos ao aprisionamento. . .	64
3.5	Elementos dos recursos humanos considerados críticos ao aprisionamento . . . .	67
4.1	1ª Aplicação - elementos dos recursos de hardware considerados críticos ao aprisionamento . . . . .	74
4.2	1ª Aplicação - elementos dos recursos de software considerados críticos ao aprisionamento . . . . .	75
4.3	1ª Aplicação - elementos dos recursos humanos considerados críticos ao aprisionamento . . . . .	78
4.4	2ª Aplicação - elementos dos recursos de hardware considerados críticos ao aprisionamento . . . . .	84
4.5	2ª Aplicação - elementos dos recursos de software considerados críticos ao aprisionamento . . . . .	85
4.6	2ª Aplicação - elementos dos recursos humanos considerados críticos ao aprisionamento . . . . .	87

# Sumário

<b>1</b>	<b>Introdução . . . . .</b>	<b>11</b>
1.1	Motivação . . . . .	13
1.2	Objetivos . . . . .	13
1.3	Organização . . . . .	14
<b>2</b>	<b>Revisão bibliográfica e conceitos básicos. . . . .</b>	<b>15</b>
2.1	Governança corporativa. . . . .	15
2.2	Governança de Tecnologia da Informação . . . . .	17
2.2.1	Definição da governança de TI . . . . .	17
2.2.2	Governança de TI como processo . . . . .	19
2.2.3	Governança de TI como estrutura. . . . .	20
2.3	Gestão e gerenciamento de TI . . . . .	21
2.4	Diferença entre a governança e a gestão de TI . . . . .	23
2.5	Infraestrutura de TI . . . . .	23
2.5.1	Recursos de hardware. . . . .	26
2.5.2	Recursos de software . . . . .	29
2.6	Aprisionamento tecnológico ( <i>lock-in</i> ) . . . . .	46
2.6.1	O ciclo do aprisionamento . . . . .	46
2.6.2	Estratégia de aprisionamento para compradores . . . . .	47
2.6.3	Custo de troca. . . . .	48
2.7	Considerações do capítulo . . . . .	49
<b>3</b>	<b>Análise do aprisionamento (<i>lock-in</i>) na gestão e infraestrutura de TI. . . . .</b>	<b>50</b>
3.1	Governança de TI limitada pelo fenômeno do aprisionamento. . . . .	50
3.2	Aprisionamento na gestão de TI . . . . .	52
3.3	Localizando o aprisionamento tecnológico na infraestrutura de TI . . . . .	53
3.3.1	Aprisionamento nos recursos de hardware . . . . .	55
3.3.2	Aprisionamento nos recursos de software . . . . .	56
3.3.3	Aprisionamento nos recursos humanos. . . . .	61
3.4	Proposta de índice de aprisionamento na infraestrutura de TI . . . . .	62
3.4.1	Parte 1 - Elementos que compõem os recursos de hardware . . . . .	62
3.4.2	Parte 2 - Elementos que compõem os recursos de software . . . . .	63
3.4.3	Parte 3 - Elementos que compõem os recursos humanos. . . . .	67

3.5	Proposta de metodologia para aplicação do índice . . . . .	70
3.6	Cálculo do índice de aprisionamento na infraestrutura de TI. . . . .	71
<b>4</b>	<b>Estudos de caso . . . . .</b>	<b>73</b>
4.1	1a Aplicação do índice . . . . .	73
4.1.1	Objetivo . . . . .	73
4.1.2	Descrição do caso. . . . .	73
4.1.3	Aplicação do índice . . . . .	74
4.1.4	Cálculo do índice de aprisionamento na aquisição . . . . .	81
4.1.5	Considerações. . . . .	82
4.2	2a Aplicação do índice . . . . .	82
4.2.1	Objetivo . . . . .	82
4.2.2	Descrição do caso. . . . .	83
4.2.3	Aplicação do índice . . . . .	83
4.2.4	Cálculo do índice de aprisionamento na aquisição . . . . .	90
4.2.5	Considerações. . . . .	91
<b>5</b>	<b>Conclusão . . . . .</b>	<b>92</b>
5.1	Trabalhos futuros . . . . .	95
	<b>Referências . . . . .</b>	<b>96</b>

# 1 Introdução

O ambiente da Tecnologia da Informação (TI) é composto por uma infraestrutura cada vez mais complexa, integrada por itens mais sofisticados e que requerem maior nível de integração tecnológica. Esse ambiente altera com enorme facilidade e velocidade os paradigmas e processos de uma organização.

Amparada por esse pujante ambiente da Tecnologia da Informação, a governança corporativa não se guia mais só por aspectos financeiros e de adequação ao mercado, passando a orientar-se também baseada em princípios como transparência, equidade, prestação de contas e responsabilidade corporativa [13], [14], [2], [15].

Como em qualquer processo evolutivo, a governança corporativa é irregular, passando por problemas que motivaram ações como a publicação do Código de Boas Práticas de Governança Corporativa, pela Comissão *Cadbury*, em 1992 e a criação, nos Estados Unidos, da legislação *Sarbanes-Oxley* (SOX), em julho de 2002, que obrigou as empresas de capital aberto a tornarem suas demonstrações financeiras mais detalhadas e transparentes para acionistas e para a sociedade em geral [15], [14], [16].

Nesse contexto, a governança de Tecnologia da Informação está relacionada à governança corporativa na medida em que se preocupa em viabilizar, por meios tecnológicos, o controle e a transparência das organizações. Dessa forma, a governança de TI deve ser considerada como um dos componentes da governança corporativa [1], [17] e [5].

Nesse ambiente da TI, envolto pela governança e pela gestão das organizações, existe um fenômeno que resulta das decisões tomadas e das ações executadas. Esse fenômeno é denominado aprisionamento tecnológico ou *lock-in*.

O aprisionamento tecnológico ou *lock-in* é a anulação do processo de mudança, levando à permanência na situação atual, que decorre dos custos envolvidos no momento em que as organizações tomam decisões pela mudança de suas estratégias, dos seus processos ou das tecnologias aplicadas [9] e [10].

É nesse momento de mudança, que os custos de troca podem levar a uma decisão de manutenção, anulando o processo de mudança, que pode ocorrer um aprisionamento à situação atual [9].

Ao observarmos a tomada das decisões fundamentais em TI, notamos que, se a arquitetura da empresa define dados e infraestruturas como plataforma estável, de forma a eles darem suporte às aplicações específicas de negócios (principalmente se a missão da empresa não for alterada durante o processo de mudança), é possível afirmar que a observância do ciclo do aprisionamento, no momento de definir os dados e a infraestrutura, facilitará o suporte às aplicações (uma vez que os custos de troca tornam-se altos, dificultando a migração para novas tecnologias). E que a dependência de determinada tecnologia diminuirá se a empresa conseguir antecipar vários ciclos futuros ao traçar determinada estratégia [9], [3], [18].

Observa-se que o aprisionamento pode ser caracterizado pela existência de um ciclo que se inicia no momento da seleção de uma marca. Quando se trata da primeira escolha, não

existe ainda preferência, mas após a opção pela primeira marca incorre-se no aprisionamento, reduzindo-se a mobilidade para realizar a próxima escolha [9].

Tendo escolhido a marca, a próxima fase é a da experimentação, na qual o usuário estará testando a marca e usufruindo das vantagens e incentivos recebidos por ela.

Sequencialmente, entra-se na fase do entrincheiramento, que ocorre quando o usuário acostuma-se com a nova marca, dando preferência a ela em relação a outras. Quanto mais tempo durar esta fase, maiores serão os custos de troca, aprofundando o aprisionamento, já que os custos de mudança tornam-se altos ao se cogitar a troca de fornecedor, mesmo para setores com custos menores para novos entrantes, como é o caso da área de TI [19].

Hess e Ricart [10] propuseram em seu estudo um conceito de graus de custos de troca que se relaciona com o a ideia de aprisionamento (*lock-in*). Esse conceito busca representar a característica dinâmica e contínua dos custos de troca, mostrando que eles podem estar sempre sendo alterados pelas decisões gerenciais.

Esses custos de troca estão no contexto dos investimentos em TI. O investimento em TI precisa mais do que um retorno mínimo (*Return on Investment* - ROI), pois a empresa necessita de um valor adicional aos negócios que decorre das mudanças organizacionais que a TI proporciona [5]. Assim, em se tratando de mudanças organizacionais, manter suas opções abertas e gerir os custos de troca desde o início irá maximizar as suas opções durante todo o ciclo do aprisionamento [20] [21].

Outro aspecto relevante é a gestão de TI ser orientada a serviços modulares e configuráveis. Essa orientação possibilita uma avaliação mais detalhada da infraestrutura de TI. Alguns autores afirmam que a infraestrutura de TI pode ser composta por diversos recursos ou partes, tais como hardware, software, redes, dados e pessoas, constituindo-se de uma parte técnica e de outra social, que se inter-relacionam de forma a se ajustar e cooperar mutuamente [22], [6].

Segundo O'Brien [6], os principais componentes e atividades dos sistemas de informação dependem dos recursos humanos (os usuários finais e os especialistas em SI), de hardware (máquina e mídia), de software (programas e procedimentos), de dados (banco de dados e base de conhecimento) e de rede (mídia de comunicações e apoio de rede) para executarem atividades de entrada, processamento, produção, armazenamento e controle que convertem recursos de dados em produtos da informação.

Entre esses recursos observa-se uma interdependência que varia de acordo com o recurso e o propósito a que cada um é designado. Segundo Herman [23], os hardwares com maior capacidade permitem softwares mais complexos e esses exigem hardwares mais poderosos e, nessa simbiose, essa dupla andou de forma acelerada, porém o hardware se transformou em *commodity* e o software está desencadeando revoluções de conceitos e de comportamentos nas pessoas [23], [9].

Um aspecto que se observa em relação ao recurso de hardware são as variações de configurações de hardware inspiradas nos supercomputadores. Esses supercomputadores passaram a ser agora “servidores redimensionáveis” na extremidade superior das linhas de produto, que começam com as estações de trabalho de mesa. Companhias orientadas para o mercado, como a *Silicon Graphics*, *Hewlett-Packard* e IBM possuem um foco muito mais amplo do que apenas o de montar o computador mais rápido do mundo, e o software do computador de mesa possui muito mais pontos de sobreposição com o do supercomputador do que costumava possuir, porque ambos são montados a partir dos mesmos microprocessadores baseados em cache [6].

Observando o recurso de hardware é possível traçar um paralelo com o tipo de aprisionamento e custo de trocas associados apresentado por Shapiro e Varian [9]. O hardware tende a

depreciar na medida que envelhece ou se torna obsoleto, sendo essa obsolescência o principal fator para algumas configurações de hardware.

Para os recursos de software, é possível sistematizá-los em sete grandes categorias que apresentam desafios contínuos, sendo: software de sistemas, software de aplicação, software de engenharia/científico, software embarcado, software para linha de produtos, aplicações web/aplicativos móveis e software de inteligência artificial [24].

Para efeito de exemplificação no recurso de software, podemos destacar a evolução histórica das linguagens de programação. Essa evolução ocorreu no sentido de diminuir a dependência do software em relação ao hardware. Esta tendência se acelerou com a criação das linguagens de quarta geração (4GLS), que são não procedurais e fáceis de utilizar. A evolução continua a crescer na medida em que avanços na tecnologia orientada a objetos, a gráficos e à inteligência artificial produzem linguagem natural e interfaces gráficas com o usuário, facilitando tanto a utilização de ferramentas como de pacotes de software de programação [7], [6].

Dessa forma, o presente trabalho pretende identificar o aprisionamento na infraestrutura de TI. Os recursos da infraestrutura serão previamente selecionados e serão utilizados para a proposição de um guia para a identificação e para a quantificação do aprisionamento, de modo a auxiliar o processo de tomada de decisão das organizações.

## 1.1 Motivação

Conforme argumentos expostos, pode-se destacar alguns fatores que motivam o presente trabalho:

- a relação entre o aprisionamento tecnológico e a tomada de decisão inerente à governança e à gestão de TI;
- os recursos da infraestrutura de TI que contribuem para o aumento do aprisionamento, considerando o processo de tomada de decisão na governança e na gestão de tecnologia da informação;
- a inexistência de um índice para identificar o grau de aprisionamento de uma dada infraestrutura de TI;
- a necessidade de identificar padrões de tecnologia fechada/aberta como parâmetro para o aprisionamento;
- o levantamento de características segundo a qualidade de software como parâmetro do aprisionamento.

## 1.2 Objetivos

O objetivo geral deste trabalho é propor subsídios para a identificação do aprisionamento na infraestrutura de TI, considerando o processo de tomada de decisão na governança e na gestão de TI.

Os objetivos específicos são:

- apresentar a relação entre os elementos da infraestrutura de TI e o fenômeno do aprisionamento;

- criar um índice para identificar e quantificar o aprisionamento na infraestrutura de TI;
- aplicar o índice criado em estudos de caso selecionados a partir do contexto da governança e da gestão da infraestrutura de TI.

### 1.3 Organização

Neste primeiro capítulo foi apresentado o contexto em que este trabalho se insere, destacando-se as relações entre a infraestrutura de TI e o aprisionamento tecnológico, as motivações e os objetivos desta dissertação.

No Capítulo 2, é apresentada uma revisão bibliográfica sobre governança corporativa, governança e gestão de TI. É apresentada ainda a conceituação pertinente ao gerenciamento de TI, abrindo para os elementos que compõem a infraestrutura de TI. Serão descritos ativos de hardware e de software com o objetivo de demonstrar as características mais relevantes para relacionar com o fenômeno do aprisionamento tecnológico. Por fim, são apresentados os conceitos sobre o aprisionamento (*lock-in*).

No Capítulo 3, são analisados os aspectos decorrentes da tomada de decisão oriunda da governança e da gestão de TI que direcionam ao aprisionamento. São apresentados os ativos da infraestrutura que mais contribuem para aumentar ou diminuir o nível do aprisionamento, extraídos da literatura mencionada no Capítulo 2. Por fim, segue a apresentação da metodologia e do índice para gerenciar o aprisionamento na infraestrutura de TI.

No Capítulo 4, são apresentados os estudos de caso para validação da metodologia e do índice sugerido no Capítulo 3. São apresentadas as conclusões da aplicação em cada estudo de caso e sugeridas algumas alterações.

No Capítulo 5, são apresentadas as conclusões e propostas de trabalhos futuros.



## 2 Revisão bibliográfica e conceitos básicos

Neste capítulo será apresentada a fundamentação teórica necessária ao acompanhamento deste trabalho, detalhando os principais termos e conceitos utilizados, visando o entendimento da relação entre o fenômeno do aprisionamento com a governança, com a gestão e com a infraestrutura de TI.

### 2.1 Governança corporativa

A governança corporativa almeja elevar o valor da organização [25], [16]. Ela ultrapassa os aspectos financeiros e a adequação ao mercado aberto de ações ao se orientar por princípios como transparência, equidade, prestação de contas e responsabilidade corporativa [26] [14].

Dito de outra forma, a governança corporativa é o sistema pelo qual as empresas e demais organizações são dirigidas, monitoradas e incentivadas. Isso envolve os relacionamentos entre sócios, conselho de administração, diretoria, órgãos de fiscalização e demais partes interessadas [14].

O principal motivador para a emergência da governança é a necessidade de equacionar o chamado "conflito de agência". Ele ocorre quando proprietários (acionistas) delegam a agentes especializados (executivos) o poder de decisão sobre sua propriedade. Desta situação podem surgir desalinhamentos de interesse, que levam exatamente ao conflito de agência ou conflito agente-principal [14]. Para Windsor [27], o conceito de conflito de agência especifica que a governança corporativa lida com as formas pelas quais os provedores financeiros das corporações asseguram o retorno sobre seus investimentos.

A Governança Corporativa é a dimensão de conformidade da Governança Empresarial, com uma perspectiva histórica da empresa, enquanto que a Governança de Negócio, que é a dimensão do desempenho, tem a perspectiva futura. O modelo a seguir mostrará que investir apenas na governança corporativa não é suficiente. A preocupação com a estratégia e a visão de futuro são imprescindíveis para a organização. As conexões entre as governanças também mostram a articulação que existe entre a conformidade e a criação de valor, enquanto que o desempenho pode levar ao rastreamento [1]. Um modelo para representação da Governança Empresarial e de suas dimensões é apresentado na Figura 2.1.

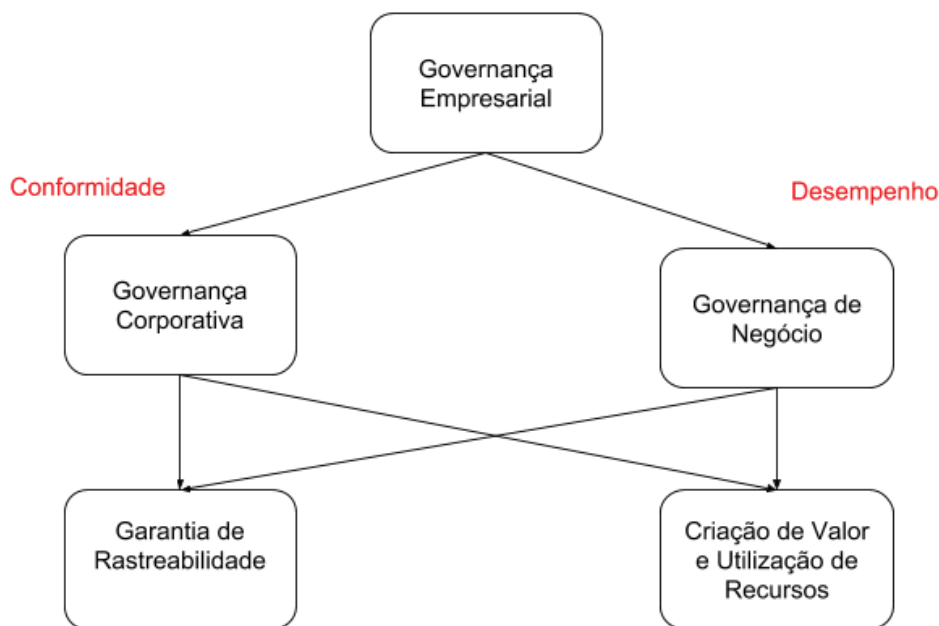


Figura 2.1: Modelo de governança empresarial. Fonte: [1]

A governança corporativa abrange aspectos da governança dos negócios de uma organização. O conjunto de responsabilidades e práticas exercido pelo alto escalão e pelo time executivo, com o objetivo de prover direção estratégica, assegura que os objetivos sejam atingidos. Isso certifica que os riscos sejam gerenciados apropriadamente e verifica se os recursos da empresa são usados com responsabilidade [1].

Os princípios para a governança corporativa das empresas não devem substituir o comando ou as iniciativas de se desenvolver melhores práticas no assunto. Observou-se que a governança corporativa deve se preocupar com o equilíbrio entre as metas econômicas e sociais, e entre as metas individuais e comuns [17], [28].

A governança corporativa, como em qualquer processo evolutivo, é irregular e inconstante [26]. Como se observa em Windsor [27], os problemas com a governança das instituições são apontados desde 1720, na Inglaterra e na Companhia das Índias Orientais.

No final dos anos 1980, uma série de problemas no mercado corporativo e financeiro motivou, em 1992, a publicação do primeiro Código de Boas Práticas de Governança Corporativa, pela Comissão Cadbury, sob comando de Sir Adrian Cadbury. Este documento influenciou os textos de manuais de governança corporativa em diversos países [14].

Nos Estados Unidos, a legislação *Sarbanes-Oxley* (SOX) foi oficializada em julho de 2002 e visava estabelecer bases para a confiança dos investidores, obrigando as empresas de capital aberto a tornarem suas demonstrações financeiras mais detalhadas e transparentes para acionistas e para a sociedade em geral, além de reforçar a independência das empresas de auditoria [29], [16].

O impacto da criação da SOX foi tão grande que ela é considerada a mais importante legislação sobre empresas de capital aberto, auditorias internas e externas e membros dos conselhos executivos desde a criação da *Security Exchange Commission* (SEC), ocorrida em 1934. Um dos efeitos da legislação *Sarbanes-Oxley* foi tornar o alto escalão (e principalmente o conselho de administração) das empresas mais atento e responsável por suas decisões e aprovações, aumentando a visibilidade sobre os processos da governança corporativa [16].

## 2.2 Governança de Tecnologia da Informação

A governança de Tecnologia da Informação (TI) está relacionada à governança corporativa e se preocupa com o controle e com a transparência das decisões em Tecnologia da Informação, sem desconsiderar mecanismos e processos para incrementar a eficácia da TI. Dessa forma, a governança de TI não deve estar apenas sincronizada com a governança corporativa, mas também deve ser considerada como um de seus componentes [15].

O fenômeno conhecido como “paradoxo da produtividade” pode ser associado à preocupação com os investimentos em TI, o que demonstra a dificuldade da Tecnologia da Informação em prover um mecanismo eficaz para medir o retorno dos investimentos [30], [31], [32].

O ambiente da Tecnologia da Informação é composto por redes, servidores e aplicativos, cada vez mais complexos, compostos por itens mais sofisticados que requerem maior nível de integração tecnológica. A crescente integração das funções administrativas e de chão de fábrica permitem a criação de sistemas do tipo *Enterprise Resource Planning* (ERP) [33]. A integração entre fornecedores e clientes, com a criação de cadeias integradas de suprimentos, e a intensificação no relacionamento com clientes, ajuda a aumentar a integração, sofisticação e complexidade do ambiente da TI [13].

A aplicação da legislação *Sarbanes-Oxley* no mercado americano ampliou a importância da governança de TI, impondo novos processos para o monitoramento das empresas. Dessa forma, para as empresas serem efetivamente inseridas no mercado globalizado se faz necessário adequações nos controles aos marcos regulatórios locais e aos códigos de governança corporativa, cada vez mais rígidos e amplos [34].

### 2.2.1 Definição da governança de TI

Para os modelos formais de governança de TI, dois fatores têm recebido atenção crescente por parte de empresas, sendo eles: o maior rigor da governança corporativa e a exigência de garantia de retorno sobre os investimentos em TI [29].

A partir dos anos 1990, as pesquisas sobre o tema da governança se intensificaram, fazendo com que a denominação "governança de TI" se tornasse mais conhecida - especialmente com os primeiros trabalhos publicados por Loh e Venkatraman em 1992 [29].

Existe uma significativa divergência entre os autores acerca da definição sobre o escopo da governança de TI. Há um grande foco na gestão operacional, explorando-se pouco os aspectos mais conceituais. Apesar de os assuntos envolvidos na governança de TI também estarem relacionados às funções da TI e existirem há muito tempo, ainda não há um corpo de conhecimento consistente e bem estabelecido da governança de TI [35], [15].

Considerando a governança de TI como um escopo de conhecimento amplo, que trata de assuntos como medição do desempenho, gestão dos recursos e controle dos riscos, a tomada de decisões, princípios, papéis e gestão do portfólio reportam trabalhos sobre estes temas desde meados dos anos 1960 [36], [3], [29], [36].

Com variados graus de sofisticação e pragmatismo, existem muitas definições sobre governança de TI. O termo "governança" é talvez o mais inadequadamente usado nos negócios, e o mesmo vale na Tecnologia da Informação, onde o termo governança de TI é frequentemente usado como sinônimo de "controle" [37].

David Musson [38] propõe três aspectos de estudo para a governança de TI: os frameworks e auditoria; a tomada de decisão; e a relação com a governança corporativa. Os frameworks da governança de TI têm por objetivo facilitar o gerenciamento e o controle da

TI, enquanto que a tomada de decisão tem foco na definição dos direitos de decisão e na atribuição de responsabilidades. Finalmente, a relação com a governança corporativa tem foco na responsabilidade do alto escalão em garantir que a TI responda aos objetivos organizacionais.

Conforme o objetivo de análise do pesquisador ou da abordagem de pesquisa, as definições apresentam focos diferentes dificultando o entendimento conceitual da governança de TI [29]. Conforme a abordagem, as definições refletem duas visões: a governança de TI como estrutura de decisão e como conjunto de processos.

A governança de TI deve buscar o compartilhamento de decisões com o negócio e o estabelecimento de regras, processos e estrutura para nortear o provimento de serviços de TI. Conforme a definição e o entendimento utilizado, a governança de TI é vista tanto como um conjunto de processos quanto como uma estrutura de decisões em TI [13].

Já no meio acadêmico, a visão predominante entende a governança de TI como estrutura de decisão e considera a forma como as responsabilidades e direitos de decisão são determinados para estimular comportamentos no uso da TI compatíveis com a estratégia da empresa. Essa visão acadêmica representa uma longa corrente de debates sobre a melhor forma (ou modo) de se tomar decisões sobre TI na organização de forma centralizada, descentralizada ou híbrida [35], [29], [34].

A governança de TI como estrutura é apresentada como um arcabouço para a tomada de decisão nas empresas, apartado das decisões propriamente ditas, e relacionam-se com o padrão de decisões em TI - os direitos de decisão e a matriz de responsabilidades [39] [18]. Esse padrão deve fomentar os chamados comportamentos desejáveis, que incorporam as crenças e a cultura da organização, extrapolando a estratégia e envolvendo as diretivas de valores, missão, princípios, rituais e estruturas de negócio e corporativas [3].

A abordagem correspondente a processos é usada para caracterizar a governança de TI como um conjunto de sistemas, processos e procedimentos modelados de forma a harmonizá-los aos da organização, garantindo o bom uso dos recursos de TI [40], [41]. Esses processos podem ser exemplificados, porém não limitados aos itens a seguir, como sendo os mecanismos de priorização de investimentos e orçamentos de TI, a estruturação dos comitês específicos e as aquisições em TI. É a eficácia destes processos que demonstram quais organizações possuem adequadas governanças em TI [41].

Uma definição de governança de TI baseada na consolidação da literatura, elaborada a partir da análise de 150 afirmativas com definições e descrições sobre o tema, aponta para o processo de decisão em TI sobre certos ativos, como hardware, software, processos, pessoal e objetivos estratégicos [42].

Os três elementos mais frequentemente associados à Governança de TI na literatura são as estruturas, os frameworks de controle e os processos. Porém, nem as estruturas em si, nem os frameworks e nem os processos são suficientes para definir a governança de TI, pois envolvem um espectro mais amplo. O estabelecimento de uma definição, que observe tal "espectro amplo" e que não limite ou restrinja o escopo conceitual, é importante para acadêmicos e gerentes obterem um entendimento comum sobre a governança de TI [41].

Além dos três elementos, são identificados mais seis constructos na literatura sobre governança de TI, sendo: alinhamento estratégico, entrega de valor, gerenciamento de desempenho, gerenciamento de riscos, políticas e procedimentos, e controle e rastreamento. Esses constructos podem ser aplicados às definições para um melhor entendimento. Porém, quatro dos seis constructos (alinhamento estratégico, entrega de valor, gerenciamento de desempenho e gerenciamento de riscos) são considerados como domínios da governança de TI [43], [41].

### 2.2.2 Governança de TI como processo

Sendo a governança de TI entendida como processo, considera-se a estratégia de negócios representada pelos direcionadores dos principais envolvidos, impulsionando os processos de TI, que precisam obter os recursos necessários à execução das responsabilidades. Essa é a visão predominante nas empresas e nas publicações de cunho não acadêmico, sendo sustentada por organizações como o *Information Systems Audit & Control Association* (Isaca) e o *Information Technology Governance Institute* (ITGI), que propõem o modelo *Control Objectives for Information and Related Technology* (Cobit) [35].

A governança de TI tem dois grandes objetivos: a entrega de valor, amparada no alinhamento dos negócios e da TI, e a mitigação de riscos, direcionada pela atribuição de responsabilidades na organização [43]. Estes dois objetivos são sustentados pelo gerenciamento adequado de recursos e pelo monitoramento do desempenho da TI. O alinhamento entre a TI e o negócio consiste em um consenso significativo sobre o escopo da governança de TI.

O alinhamento entre a TI e os negócios ou o alinhamento estratégico da TI, focaliza o grau em que os investimentos em TI estão em sincronia com os objetivos estratégicos. Em empresas com ativos significativos de TI, esse alinhamento se torna relevante pois visa um desempenho superior nos resultados oriundos da sinergia entre os recursos de TI e os objetivos da empresa. O alinhamento é considerado quase como sinônimo de estratégia de TI, entretanto abarca também a integração entre operações atuais da empresa com as da TI [43].

Considerado um tema complexo e multifacetado, o alinhamento é um estado difícil de ser atingido, pois os objetivos da organização mudam muito rapidamente [44]. O alinhamento está relacionado à eficácia, que por sua vez está relacionada à estratégia [45], [43]. A estratégia do negócio deve ser amparada pela estratégia da TI, nas dimensões futura e presente, externa e interna, requerendo integração entre estratégia, processos e operações [45].

Na literatura, são inúmeros os estudos que procuram identificar as variáveis mais relevantes ao alinhamento estratégico e propor modelos conceituais para avaliar o nível de maturidade do alinhamento. Analisando 45 trabalhos sobre alinhamento estratégico, publicados entre 1987 e 2007, [34] foram encontrados vários modelos conceituais de referência, com destaque para o Modelo de Alinhamento Estratégico de [2].

O Modelo de Alinhamento Estratégico – *Strategic Alignment Model* (SAM) – de [2], é um dos primeiros modelos e propõe o alinhamento em função de duas dimensões, o ajuste estratégico e a integração funcional, conforme a figura 2.2 abaixo:

O ajuste estratégico é composto pela estratégia de negócio (escolhas para definir o posicionamento da empresa no mercado), enquanto que a integração funcional permite à organização alinhar seus processos às variáveis internas e externas. É fundamental na busca de uma boa administração da TI o balanceamento das decisões relacionadas à estratégia de negócio, estratégia de TI, infraestrutura organizacional e de processos e infraestrutura de sistemas de informação e processo [46], [45], [34].

A preocupação em demonstrar boa Governança Corporativa para acionistas, alto escalão e demais interessados nas empresas tem reforçado as práticas de gerenciamento de riscos, que devem cuidar da proteção dos ativos de TI e dar recuperação dos negócios em caso de desastres [44].

Segundo Buckby [34], a literatura sobre gerenciamento de riscos intensificou-se a partir de 2003, impulsionada pelos problemas, fraudes e falta de transparência da Governança Corporativa em grandes organizações. Os estudos focam em três áreas de estudo: a identificação, o gerenciamento e a medição dos riscos.



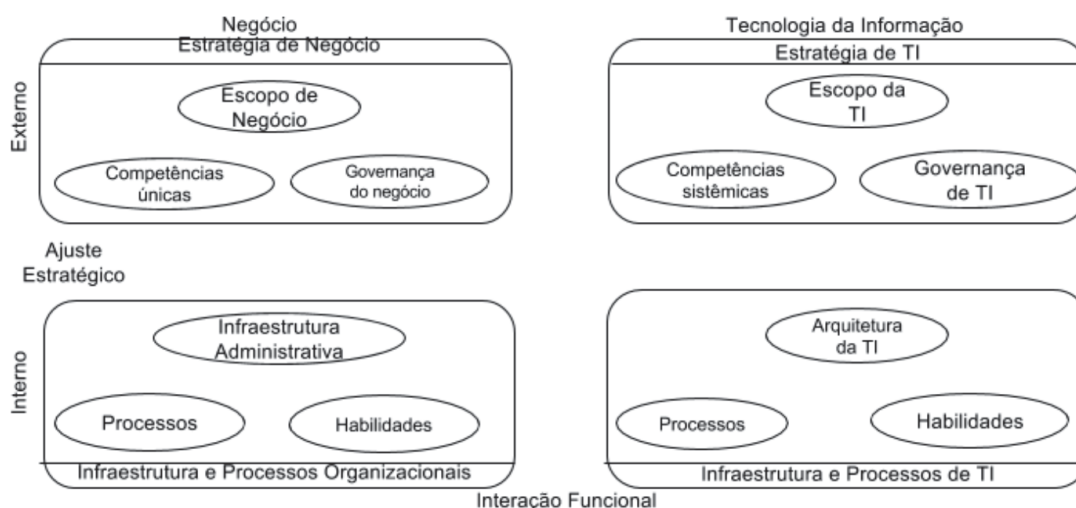


Figura 2.2: Modelo de alinhamento estratégico. Fonte: [2]

### 2.2.3 Governança de TI como estrutura

Os trabalhos relacionados às estruturas de tomada de decisão representam a visão dominante da TI e remontam à década de 1960 [47]. Referem-se à forma como as responsabilidades e direitos de decisão são determinados para estimular comportamentos no uso da TI compatíveis com a estratégia da empresa [48], [49], [40], [50], [3], [36], [3], [18], [29].

A governança de TI como uma estrutura busca atender a gestão e o fornecimento dos serviços de TI e está concentrada sobre a autoridade e o local da tomada de decisão [15]. A Governança de TI deve se concentrar na alocação, distribuição e padrões de responsabilidade gerencial que afetam como os recursos de TI serão aplicados e posteriormente implantados [51].

A identificação das decisões fundamentais em TI, a definição de quem deve tomá-las e o aparato necessário para que boas decisões sejam tomadas são as três questões centrais na tomada de decisão [3].

Estudos iniciais apresentavam como decisões fundamentais da TI o gerenciamento da infraestrutura, a gestão do uso e o gerenciamento de projetos [49], [40].

As decisões sobre a arquitetura de TI convertem os princípios de TI em requisitos de integração e padronização e, então, delineiam um guia técnico para prover as capacidades necessárias. As decisões relativas aos investimentos e à priorização da TI mobilizam recursos para converter princípios em sistemas [5].

Decisões sobre infraestrutura e aplicações podem fluir de cima para baixo (abordagem top-down) - dos princípios, da arquitetura e dos critérios de investimento. Nesse caso, a infraestrutura gera as capacidades necessárias de TI e as aplicações fazem uso dessas capacidades. Com a mesma frequência, necessidades e oportunidades de negócio identificam a necessidade de aplicações de TI, que “borbulham” da base (abordagem bottom-up) para gerar novos requisitos de infraestrutura. Por fim, as decisões de investimento selecionam e financiam as iniciativas de infraestrutura e aplicações, que implementam uma arquitetura projetada para incorporar os princípios de TI e em última instância os princípios do negócio [5].

Posteriormente, estas decisões foram revistas e ampliadas a partir de uma compilação de vários trabalhos [3], [18], expondo a interconexão entre as decisões, conforme Figura 2.3

<b>Princípios de TI</b> Definições de alto nível sobre o papel da TI nos negócios		
<b>Arquitetura de TI</b> Organização lógica de dados, sistemas e infraestrutura, organizada em políticas, relacionamentos e diretrizes tecnológicas, para obter a integração e a padronização funcional e técnica necessária ao negócio	<b>Estrutura da Infraestrutura de TI</b> Serviços de TI coordenados de forma centralizada e compartilhada, que proveem a sustentação para a capacidade de TI da empresa	<b>Investimentos e Priorização em TI</b> Decisões sobre quanto e em quais projetos investir em TI, bem como esquema de aprovações de projetos e métodos de justificativas
	<b>Necessidade Específicas do Negócio</b> Especificações das necessidades de negócios para aplicações de TI, a serem desenvolvidas internamente ou compradas no mercado	

Figura 2.3: Decisões fundamentais em TI. Fonte: [3]

Segundo Weill e Ross [3], os princípios da TI também devem definir os comportamentos desejáveis tanto para profissionais quanto para usuários da TI, esclarecendo sobre pelo menos três expectativas organizacionais para a TI: o modelo operacional desejado pela empresa, como a TI suporta o modelo operacional desejado e como a TI é financiada.

A maioria das arquiteturas empresariais demonstra a infraestrutura, os dados e as aplicações, e captura a organização lógica nas políticas e escolhas técnicas. Assim, a arquitetura de uma empresa define dados e infraestrutura como uma plataforma estável, que dá suporte a aplicações específicas de negócios, e, portanto, mais sujeitas a mudanças. Desde que a empresa não mude sua missão básica, a infraestrutura de TI suporta as aplicações de negócios [3].

Prover a infraestrutura de TI adequada significa boa relação custo-benefício que capacite a empresa a adotar rapidamente novas aplicações de negócio. Determinar onde os serviços locais de infraestrutura devem ser posicionados, como financiá-los e ratear seus custos entre as unidades de negócio, quando devem ser atualizados e se cabe ou não terceirizá-los são decisões essenciais de infraestrutura [5].

Decisões sobre necessidades de aplicações de negócio envolvem a identificação de mudanças nos sistemas e processos capazes de trazer benefícios significativos para a empresa. Considerando os recursos finitos da TI, novos aplicativos exigem o atendimento não só de um retorno mínimo sobre o investimento (*Return on Investment* – ROI), como também uma contribuição para o valor estratégico para empresa; dessa forma, o valor para os negócios decorre das mudanças organizacionais que a TI proporciona [5].

## 2.3 Gestão e gerenciamento de TI

A área de TI é responsável por prover e gerenciar os serviços de manutenção de hardware, software, dados e redes, usando várias competências profissionais: analistas de sistemas, programadores, arquitetos de informação, consultores e gerentes de projetos, entre outros. O principal executivo, o Diretor Executivo de Tecnologia da Informação (CIO), é o responsável maior pelo gerenciamento da função TI na organização [52].

Nesse enquadramento, a gestão é a presença de algum sistema de autoridade e administração, personificada por um ou vários gerentes numa hierarquia, para "amarrar" todos os esforços conjuntamente. Esse "amarrar" de esforços requer várias funções, e sua realização leva ao gerenciamento propriamente dito [52].

A maior parte do trabalho gerencial é criar e seguir uma agenda adequada ao monitoramento operacional do negócio, visando construir certezas - principalmente através do controle financeiro. Entretanto, a sobrevivência da empresa demanda um empenho tanto no gerenciamento operacional quanto no gerenciamento estratégico [4].

Para Steele [4] o sucesso no gerenciamento da tecnologia envolve um balanceamento contínuo de alternativas contraditórias. A contradição ocorre entre as preocupações e características do gerenciamento estratégico - centrado nas mudanças necessárias à sobrevivência da empresa, e do gerenciamento operacional - focado na eficiência de curto prazo conforme a Figura 2.4.



Figura 2.4: Características e preocupações do gerenciamento operacional e estratégico. Fonte: [4]

O gerenciamento estratégico contempla a implantação de modificações na maneira como a empresa compete e sobrevive no mercado, controlando ações e comportamentos necessários para implantar as mudanças [4].

No âmbito operacional, o gerenciamento dos serviços de TI é o conjunto de processos que cooperam para assegurar a qualidade aos serviços de TI, de acordo com os níveis acordados com o cliente [53]. Envolve a escolha dos programas de tecnologia, a avaliação de riscos e incerteza, o gerenciamento de projetos e a transferência e implantação da tecnologia [4].

Nas funções de gerenciamento, a TI tem evoluído ao longo do tempo, transformando-se de um provedor de tecnologia em um provedor de serviços, e mais recentemente, num parceiro estratégico [53]. A TI como uma provedora de tecnologia tem foco na melhoria do gerenciamento da infraestrutura e deve garantir o melhor retorno sobre os investimentos em ativos computacionais e nos dados gerenciados.

Como provedora de serviços, a TI está ativamente envolvida com os usuários e tem como principais responsabilidades o planejamento e a entrega de serviços que possam atender às



necessidades de seus clientes, conforme os requisitos de disponibilidade, desempenho, segurança, qualidade e custo, e o gerenciamento de acordos de nível de serviço (Service Level Agreement - SLA), tanto com provedores internos quanto externos.

Nesses estágios (provedor de serviço e provedor de tecnologia), o gerenciamento da TI se caracteriza por: foco na eficiência operacional, possibilidade de se descolar dos negócios (sem prejuízo notável à continuidade da empresa), forte disciplina de custos e orçamentos baseados em parâmetros de mercado. A TI das funções de gerenciamento é classificada como despesa a se controlar e os gerentes têm competências predominantemente técnicas [44], [53].

O gerenciamento dos recursos trata da otimização do conhecimento e da infraestrutura, almejando o alto desempenho: otimização dos investimentos, dos usos e da alocação dos recursos da TI. Os recursos da TI podem ser exemplificados por pessoas, aplicativos, equipamentos e dados. Estes ativos são de difícil gerenciamento e se transformam continuamente em função da natureza da TI, dos avanços tecnológicos e das mudanças nos requisitos de negócios [43].

A monitoração do desempenho da TI trata do rastreamento da entrega dos projetos e do monitoramento dos serviços de TI. Neste contexto, os indicadores de medição de desempenho assumem importância fundamental: sua escolha determina se a TI está caminhando corretamente na direção estabelecida pelos negócios [43].

## 2.4 Diferença entre a governança e a gestão de TI

É considerado sinal de má governança corporativa delegar a governança de TI ao nível dos gerentes de TI. Tecnologia da Informação é parte integrante dos negócios e governança da TI é parte integrante da governança corporativa [15].

A caracterização da governança de TI como uma forma de gerenciamento de TI é um mito, apesar de haver diferenças tênues entre as duas [15]. As diferenças têm implicações importantes na modelagem da governança de TI: o escopo do gerenciamento contempla a eficiência e a eficácia da provisão de serviços e produtos de TI internamente à organização, bem como no gerenciamento das operações de TI.

A governança de TI, por outro lado, é mais ampla, e se concentra na viabilização e na transformação da TI para atender às necessidades atuais e futuras do negócio (foco interno) e dos clientes (foco externo) [15].

Enquanto que partes ou processos inteiros da gestão ou da provisão de produtos e serviços de TI podem ser subcontratados ("terceirizados"), principalmente os considerados como básicos (commodities), a governança de TI é específica para cada organização, define os controles e direcionamentos para a TI e não pode ser delegada ao mercado [15].

Nessa abordagem, cabe à governança da TI a aproximação e a sincronização com os objetivos da governança corporativa, os quais são usados na criação dos objetivos e métricas necessárias para o adequado gerenciamento da TI. Segundo Sallé [53], a TI deve ser a integração do gerenciamento da TI - que é orientado a serviços - com a governança da TI - que deve alinhar a TI com os objetivos da governança corporativa.

## 2.5 Infraestrutura de TI

Considerando que o escopo do gerenciamento contempla a eficiência e a eficácia da provisão de serviços e produtos de TI internamente à organização, bem como no gerenciamento das operações de TI [15], faz-se necessário identificar os componentes da infraestrutura de TI.

A infraestrutura dá suporte a demandas internas da organização (aplicações locais de TI), bem como realiza a interface com outras organizações externas (clientes, parceiros de negócios, infraestruturas públicas). Essa infraestrutura de TI possui os seguintes componentes conforme a representação na Figura 2.5 [5].



Figura 2.5: Componentes da infraestrutura de TI. Fonte: [5]

No total, foram identifica 70 serviços de infraestrutura de TI. Nesse sentido, cada um dos componentes representa um subconjunto desses serviços. Os componentes foram agrupados em duas camadas: física e gestão [54] [5]. A camada física compreende os seguintes componentes:

**Canais eletrônicos integrados** – serviços de comunicação com organizações externas através de diferentes canais de acesso;

**Segurança e risco** – serviços de segurança e gestão de risco por meio de tecnologias como firewalls, políticas de controle de acesso remoto, uso de senhas, bem como o planejamento e a recuperação a desastres;

**Comunicações** – serviços de comunicação dentro da própria organização e entre clientes e parceiros. Ocorrem através de interações eletrônicas por voz, vídeo, intranet, entre outras;

**Administração de dados** – gestão dos dados de clientes, produtos e processos. Serviços garantem que informações críticas estejam disponíveis a pessoas e a aplicações sempre que for necessário;

**Aplicações de infraestrutura** – serviços utilizados por toda a organização (contabilidade, recursos humanos, orçamento, etc.). Por exemplo, o Enterprise Resource Planning (ERP) e o Customer Relationship Management (CRM). Possuem estreita relação com a “administração de dados”;

**Administração das instalações de TI** – serviços de gerenciamento da camada física. Por exemplo, processamento em larga-escala, ambientes de desenvolvimento, gerenciamento das aplicações, etc.;

A camada de gestão compreende componentes que tratam da gestão de toda a infraestrutura de TI:

**Administração de TI** – serviços de coordenação da infraestrutura integrada e gerenciamento de sua relação com as unidades de negócio. Inclui planejamento de sistemas de informação,

gestão de projetos, acordos de nível de serviço (service level agreements – SLAs) e negociação com fornecedores;

**Arquitetura e padrões de TI** – políticas no uso da TI em toda a organização. Os padrões são importantes ao levar em consideração a comunicação entre os diferentes atores na cadeia de valor. Os serviços provêm de um plano de migração para outros padrões técnicos. Estão em constante desenvolvimento;

**Educação e treinamento em TI** – frequentemente esquecida, trata-se da educação em tecnologias e sistemas da organização, bem como o treinamento dos gestores sobre como investir e usar a TI para criar valor organizacional. Organizações que gastam mais em treinamento conseguem obter melhor desempenho;

**Pesquisa e desenvolvimento de TI** – interagindo com os grupos “administração de TI” e “arquiteturas e padrões de TI”, busca novos meios de usar a TI para agregar valor.

As pesquisas demonstram uma correlação positiva entre a rapidez de implantação de novas iniciativas de negócio e a infraestrutura de TI. As organizações que administram a infraestrutura como um ativo e realizam investimentos com frequência, geralmente conseguem melhores resultados [5].

Ao envolver tecnologia, processos e pessoas, uma das características principais da infraestrutura de TI é ser orientada a serviços modulares e configuráveis [55]. A visão da infraestrutura como serviço ajuda os gestores a determinarem o valor da TI. Auxilia, por exemplo, na comparação de preços entre fornecedores. Do ponto de vista do fornecedor também é interessante uma vez que padroniza os serviços oferecidos independente das mudanças tecnológicas [54].

Ao observarmos a literatura acerca dos sistemas de informação, notamos também uma definição que abrange aspectos da infraestrutura. O’Brien [6] define especificamente que sistema de informação é um conjunto organizado de pessoas, hardware, software, redes de comunicações e recursos de dados que coleta, transforma e dissemina informações em uma organização. No contexto deste trabalho, sistema de informação estará sempre associado ao uso amplo e irrestrito das tecnologias de informação e comunicação para o armazenamento e recuperação da informação.

Para Rezende [22] os sistemas podem ser compostos por diversas partes, tais como hardware, software, dados e pessoas, constituindo-se de uma parte técnica e outra social que se interrelacionam de forma a se ajustar e cooperar mutuamente. Herman [23], defende que hardwares com maior capacidade permitem softwares mais complexos e que softwares mais complexos exigem hardwares mais poderosos. Nessa simbiose, essa dupla andou de forma acelerada, porém o hardware se transformou em commodity, o software está desencadeando revoluções de conceitos e comportamento nas pessoas.

Segundo O’Brien [6] os principais componentes e atividades de um sistemas de informação dependem dos recursos humanos (os usuários finais e os especialistas em SI), de hardware (máquina e mídia), software (programas e procedimentos), dados (banco de dados e base de conhecimento) e rede (mídia de comunicações e apoio de rede) para executar atividades de entrada, processamento, produção, armazenamento e controle que convertem recursos de dados em produtos da informação.



Figura 2.6: Os componentes de um sistema de informação. Fonte: [6]

Esse modelo de componentes destaca as relações entre suas atividades. Ele fornece uma estrutura referencial que enfatiza quatro conceitos principais que podem ser aplicados a todos os tipos de sistemas de informação:

- pessoas, hardware, software, dados e redes são os cinco recursos básicos dos sistema de informação;
- os recursos humanos consistem em usuários finais e especialistas em SI; os recursos de hardware consistem em maquinais e mídias; os recursos de software consistem em programas e procedimentos; os recursos de dados consistem em bancos de dados e bases de conhecimento; e os recursos de redes consistem em mídias e redes de comunicações;
- os recursos de dados são transformados por atividades de processamento de informação em uma diversidade de produtos de informação para usuários finais;
- processamento de informação consiste em atividades de entrada, processamento, saída, armazenamento e controle.

### 2.5.1 Recursos de hardware

Conforme O'Brien [6] o conceito de hardware compreende todos os dispositivos físicos e equipamentos utilizados no processamento de informações. Elementos esses que vão de computadores e outros equipamentos até os vários tipos de mídias de armazenamento de registro de dados como exemplo discos magnéticos e até folhas de papel.

O hardware consiste na tecnologia para processamento computacional, armazenamento, entrada e saída de dados. Esse componente inclui grandes mainframes, servidores, computadores de médio porte, computadores pessoais e laptops, assistentes digitais pessoais (PADs) de mão e dispositivos móveis que dão acesso a dados corporativos e à internet. Inclui, também, equipamentos para reunir e registrar dados, meios físicos para armazenar os dados e dispositivos para a saída da informação processada [56].

Um computador é mais do que um conjunto de dispositivos eletrônicos realizando uma multiplicidade de tarefas de processamento de informações. É um sistema, uma combinação de componentes inter-relacionados que desempenham as funções básicas do sistema, ou seja,

entrada, processamento, saída, armazenamento e controle, fornecendo assim aos usuários finais uma poderosa ferramenta de processamento de informações. Compreender o computador como um sistema é vital para o uso eficaz do gerenciamento de computadores [6].

Os microcomputadores são a categoria mais importante de sistemas de computadores para usuários finais. Embora costume ser chamado de computador pessoal ou PC, um microcomputador é muito mais do que um pequeno computador para uso individual. O poder de computação dos microcomputadores agora excede o dos mainframes das gerações anteriores - conseguem isso por uma fração do custo destes computadores. Dessa forma, tornam-se poderosas estações de trabalho profissionais interconectadas para os usuários finais nas organizações [6]. Um computador pessoal ou PC pode ser de mesa ou laptop, um assistente digital pessoal, como um BlackBerry, iPhone ou Palm, ou outros modelos de celulares. Se a utilização for com engenharia ou projetos avançados, que exigem gráficos ou computacionais poderosos, talvez uma estação de trabalho (workstation), que também se encaixa na categoria de computador de mesa, mas possui capacidade de processamento matemático e gráfico superior a um PC [56].

Para Brooksheat [7], o conjunto de circuitos de computador que controla a manipulação de dados é chamado de Unidade Central de Processamentos ou CPU. Uma CPU consiste em três partes conforme a figura 2.7: a unidade lógica e aritmética, que contém os circuitos que realizam operações sobre dados (como adição e subtração), a unidade de controle, que contém os circuitos para a coordenação das atividades de máquina; e a unidade de registro, que contém células de armazenamento de dados (similar às células da memória principal, chamadas de registradores, usadas para o armazenamento temporário das informações dentro da CPU).

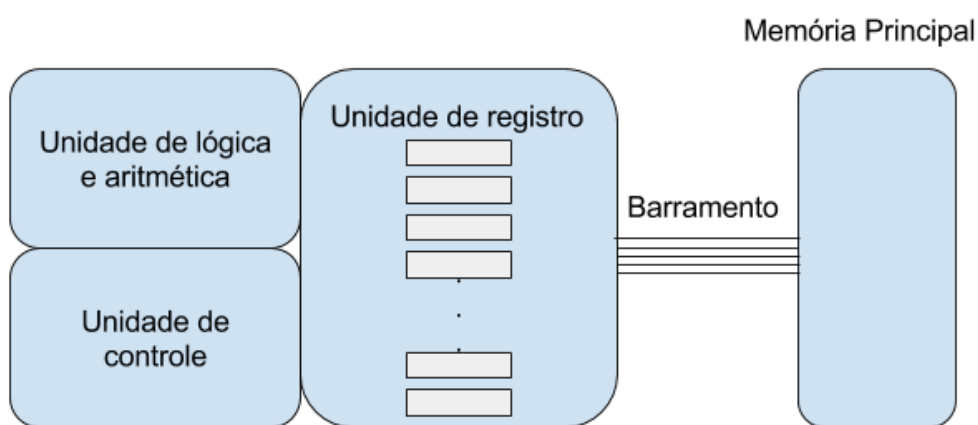


Figura 2.7: CPU e memória principal conectadas via barramento. Fonte: [7]

Outra categoria de computadores é a dos mainframes. Os mainframes apareceram pela primeira vez em meados da década de 1960 e, desde então, são usados por grandes bancos, companhias de seguros, corretoras de ações, sistemas de reservas aéreas, órgãos públicos e muitos outros tipos de organização para controlar centenas de milhares, até milhões, de registros e transações. Um mainframe é um computador de alto desempenho e grande capacidade, capaz de processar gigantescas quantidades de dados com extrema velocidade. A IBM, líder do segmento, remodelou seus sistemas de mainframe de maneira que possam ser usados como gigantescos servidores para sites corporativos e redes integradas de larga escala [56].

Os mainframes são sistemas grandes, rápidos e potentes. Possuem alta capacidade de processamento - centenas de milhões de instruções por segundo (mips) - e de milhares de gigabytes. Além disso como resultado nova tecnologia CMOS (refrigeração a ar e não a água), os mainframes foram drasticamente compactados nos últimos anos, reduzindo suas necessidades



de ar-condicionado, consumo de energia elétrica, requisitos de espaço e, com isso, seus custos de aquisição e operação [6].

Existe ainda a categoria do tipo servidor que são otimizados especialmente para suportar uma rede de computadores, permitindo aos usuários compartilhar arquivos, software, dispositivos periféricos (impressoras, por exemplo) ou outros recursos. Os servidores tornam-se componentes importantes da infraestrutura de TI das organizações. Com adição de software especial, podem ser especialmente projetados para fornecer plataformas de comércio eletrônico, apresentar páginas web, processar transações de compra e venda ou permutar dados com outros sistemas internos da organização [56].

A proliferação dos data warehouses e aplicações afins, como data mining e processamento analítico online (Olap), estão obrigando as instalações de TI a necessitarem de configurações de servidores cada vez mais poderosas. Do mesmo modo, as aplicações baseadas na internet, com servidores de rede e comércio eletrônico, estão obrigando os gerentes de TI a adicionar novas aplicações (gerenciais) e ampliar a velocidade de processamento como a capacidade de armazenamento e outras aplicações (gerenciais) que alimentam o crescimento de sofisticados servidores [6].

Os computadores de médio porte incluindo microcomputadores e sofisticados servidores de rede, são sistemas para múltiplos usuários que podem gerenciar redes de PCs e terminais. Embora não tão potentes, são mais baratos que os sistemas de mainframes em termos de compra, operação e manutenção, por isso, atendem às necessidades de computação de muitas organizações. Os computadores de médio porte se tornaram populares nas empresas como poderosos servidores de rede para ajudar a gerenciar sites na Internet, intranets e extranets e redes cliente/servidor [6].

Os supercomputadores utilizam arquiteturas de processamento paralelo de microprocessadores interconectados (que podem executar mais instruções em paralelo ao mesmo tempo). Eles conseguem fazer cálculos aritméticos a velocidades de bilhões de operações com ponto flutuante por segundo (gigaflops). Contudo, já estão disponíveis supercomputadores teraflop (um trilhão de operações com ponto flutuante por segundo), que utilizam projetos avançados de gigantescos processamentos paralelos de milhares de microprocessadores interconectados, com as pesquisas já estando no petaflop (mil teraflops) [6].

Os supercomputadores passaram a ser agora “servidores redimensionáveis” na extremidade superior das linhas de produto que começam com as estações de trabalho de mesa. Companhias orientadas para o mercado, como a *Silicon Graphics*, *Hewlett-Packard* e *IBM* possuem um foco muito mais amplo do que apenas o de montar o computador mais rápido do mundo, e o software do computador de mesa possui muito mais pontos de sobreposição com o do supercomputador do que costumava possuir, porque ambos são montados a partir dos mesmos microprocessadores baseados em cache [6].

Conforme Pitanga [57], na sua forma mais básica, um cluster é um sistema que compreende dois ou mais computadores ou sistemas (denominados nodos) que trabalham em conjunto para executar aplicações ou realizar outras tarefas, de tal forma que os usuários do agrupamento de máquinas tenham a impressão de que somente um único sistema responde para eles, criando a ilusão de um recurso único (computador virtual) conforme a Figura 2.8.

Segundo Hennessy [58], em vez de contar com máquinas personalizadas e redes personalizadas para construir máquinas maciçamente paralelas, a introdução de switches como parte da tecnologia de rede significou que a alta largura da banda de rede e o aumento de escala das redes estavam disponíveis a partir de componentes prontos. Quando combinada com o uso de computadores desktop e discos como dispositivos de computação e armazenamento, poderia ser criada uma infraestrutura de computação muito menos dispendiosa, capaz de resolver problemas

muito grandes. Além disso, pela natureza de seus componentes, os clusters têm muito mais facilidade para aumentar de escala e para isolar falhas.

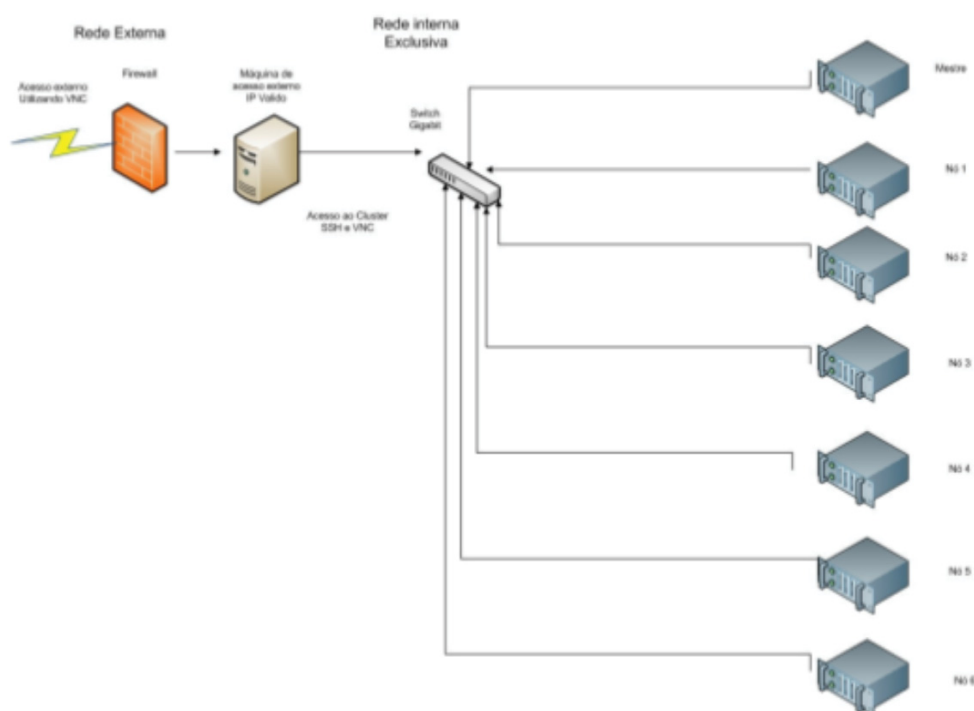


Figura 2.8: Esquema do cluster com acesso externo. Fonte:[8]

## 2.5.2 Recursos de software

Segundo Pressman [24], um software consiste em: (1) instruções (programas de computador) que, quando executadas, fornecem características, funções e desempenho desejados; (2) estrutura de dados que possibilitam aos programas manipular informações adequadamente; e (3) informação descritiva, tanto na forma impressa quanto na virtual, sobre a operação e o uso dos programas. Software é mais um elemento lógico do que físico. Portanto, o software tem uma característica fundamental que o torna consideravelmente diferente do hardware: o software não se “desgasta” [24].

Para O’Brien [6] o software refere-se a todos os conjuntos de instruções de processamento da informação. É a parte lógica que dará instruções para a execução do hardware.

O software pode ser dividido em dois tipos: os softwares de sistema, que administram os recursos e as atividades dos computadores e os softwares aplicativos, que são desenvolvidos com um objetivo específico solicitado pelo usuário final [59].

Dentre os softwares de sistema temos o sistema operacional, que gerencia e controla as atividades do computador. Ele é responsável por prover recursos do hardware para as aplicações, alocar memória, controlar os dispositivos de entrada e saída e a execução de diversas tarefas realizadas pelo computador. Outro software de sistema é caracterizado por converter a linguagem de programação para linguagem de máquina, bem como programas utilitários que executam tarefas comuns, como cálculos e classificações [59].

De acordo com Rezende [22], o sistema operacional administra o hardware, o software e seus periféricos, determinando os recursos computacionais necessários para a realização de determinada tarefa. Os sistemas operacionais podem ser divididos em proprietários ou livres.

Já o software aplicativo, que pode ser desenvolvido pela empresa que o utilizará ou adquirido de terceiros, caracteriza-se por uma aplicação utilizada pelos usuários finais, desenvolvida com o objetivo de resolução de um problema específico, utilizando os recursos de hardware por meio do sistema operacional. Dentre os softwares aplicativos mais utilizados pode-se citar os editores de texto, planilhas eletrônicas, gestão de dados, recursos gráficos de apresentação e navegadores web [59].

Os softwares aplicativos ou de escritório compreendem editores de texto, planilhas eletrônicas, softwares de apresentação e navegadores, sendo utilizados pelos usuários finais para resolução de problemas corriqueiros. Os softwares utilitários têm com função complementar os de escritório e são compostos por softwares como antivírus, compactadores, desfragmentadores e softwares de cópia [22].

Outra forma de sistematizar o software é em sete grandes categorias que apresentam contínuos desafios [24]:

- **software de sistema** - conjunto de programas feito para atender a outros programas. Certos softwares de sistema (por exemplo, compiladores, editores e utilitários para gerenciamento de arquivos) processam estruturas de informação complexa, porém determinadas. Outras aplicações de sistemas (por exemplo, componentes de sistema operacional, drivers, software de rede, processadores de telecomunicações) processam dados amplamente indeterminados.
- **software de aplicação** - Programas independentes que solucionam uma necessidade específica de negócio. Aplicações nessa área processam dados comerciais ou técnicos de uma forma que facilite operações comerciais ou tomadas de decisão administrativas/-técnicas.
- **software de engenharia/científico** - uma ampla variedade de programas de “cálculo em massa” que abrangem astronomia, vulcanologia, análise de estresse automotivo, dinâmica orbital, projeto auxiliado por computador, biologia molecular, análise genética e meteorologia, entre outros.
- **software embarcado** - residente num produto ou sistema e utilizado para implementar e controlar características e funções para o usuário e para o próprio sistema. Executa funções limitadas e específicas (por exemplo, controle do painel de um forno micro-ondas) ou fornece função significativa e capacidade de controle (por exemplo, funções digitais de automóveis, tal como controle do nível de combustível, painéis de controle e sistemas de freio).
- **software para linha de produtos** - projetado para prover capacidade específica de utilização por muitos clientes diferentes. Software para linha de produtos pode se concentrar em um mercado hermético e limitado (por exemplo, produtos de controle de inventário) ou lidar com consumidor de massa.
- **aplicações web/aplicativos móveis** - esta categoria de software voltada às redes abrange uma ampla variedade de aplicações, contemplando aplicativos voltados para navegadores e software residentes em dispositivos móveis.
- **software de inteligência artificial** - faz uso de algoritmos não numéricos para solucionar problemas complexos que não são passíveis de computação ou de análise direta. Aplicações nessa área incluem: robótica, sistemas especialistas, reconhecimento de padrões (de imagem e voz), redes neurais artificiais, prova de teoremas e jogos.



Na figura 2.9 é sugerido uma visão geral das principais categorias de software aplicativo e de sistemas em uma ilustração conceitual [6]:

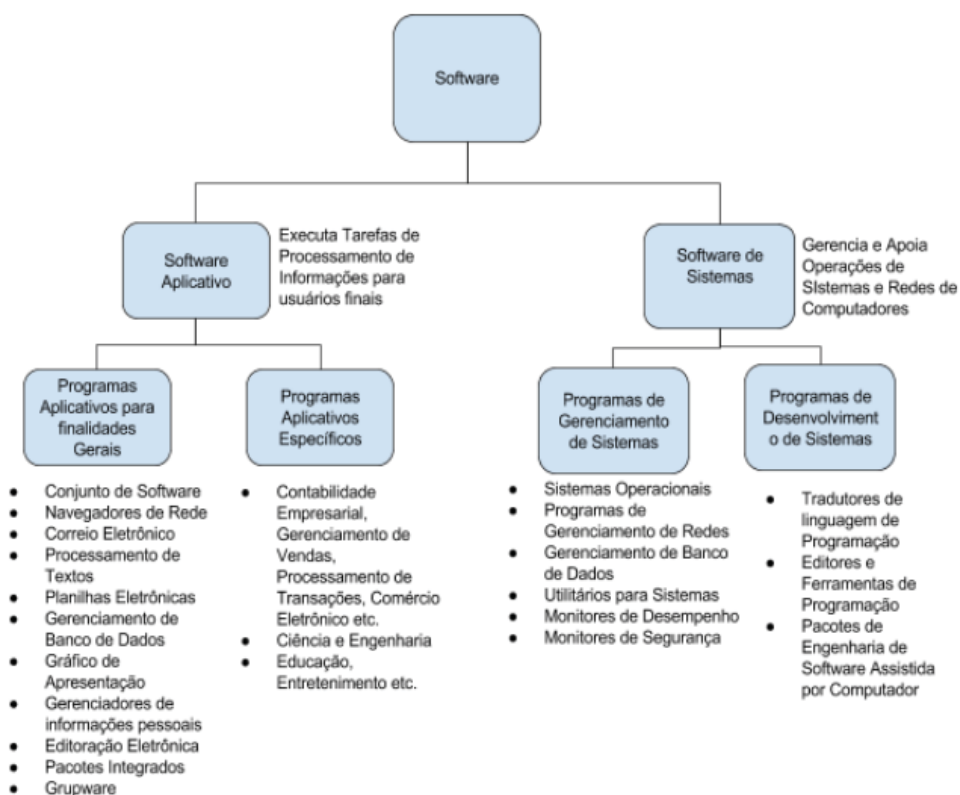


Figura 2.9: Uma visão geral do software. Fonte:[6]

Mesmo com o modelo conceitual apresentado acima, podem ser encontrados outros tipos de software, pois os mesmos variam conforme os tipos de computadores e redes utilizados e de quais tarefas específicas serão executadas [6].

A figura 2.9 mostrou que o software aplicativo inclui uma série de programas que podem ser subdivididos nas categorias de finalidades gerais e de aplicações específicas. Milhares de pacotes de softwares específicos são encontrados para apoiar aplicações específicas de usuários finais em empresas, outros campos e usuários finais - por exemplo o gerenciamento de finanças pessoais, entretenimento e educacionais [6].

Já os programas aplicativos de finalidades gerais são programas que executam trabalhos comuns de processamentos de informações para usuários finais. Os programas de processamento de textos, planilhas eletrônicas, gerenciamento de banco de dados e programas gráficos são populares entre usuários de microcomputadores para residências, educação, negócios, ciências e muitos outros processos. Por aumentarem significativamente a produtividade dos usuários finais, às vezes são conhecidos como pacotes de produtividade. Outros exemplos incluem navegadores de redes, correio eletrônico (e-mail) e o groupware, que ajuda a comunicação e colaboração entre grupos e equipes de trabalho [6].

O software de sistemas, demonstrado na Figura 2.9, consiste em programas que gerenciam e apoiam um sistema de computador e suas atividades de processamento de informações. O software de sistemas pode ser agrupado em duas categorias principais [6]:

- **programas de gerenciamento de sistemas** que gerenciam recursos de hardware, software, rede e dados do sistema de computador durante a execução dos vários trabalhos de processamento de informações dos usuários;

- **programas de desenvolvimento de sistemas** que ajudam os usuários a desenvolverem programas e procedimentos de sistemas, a exemplo dos tradutores e editores de linguagem de programação.

### *Linguagem de programação*

O desenvolvimento de sistemas complexos de software, como sistemas operacionais, software de rede e a vasta quantidade de aplicativos de software disponível hoje, provavelmente seria impossível se os humanos fossem forçados a escrever programas em linguagens de máquina. Consequentemente, linguagens de programação similares ao nosso pseudocódigo têm sido desenvolvidas buscando permitir que os algoritmos sejam expressos em um formato palatável aos humanos e facilmente convertidos em instruções de linguagem de máquina [7].

Programas chamados montadores foram desenvolvidos para converter expressões mnemônicas em instruções de linguagem de máquina. Logo, ao invés de ser forçado a desenvolver um programa diretamente em linguagem de máquina, um humano poderia desenvolver um programa em forma mnemônica e, então, convertê-lo em linguagem de máquina por meio do montador [7].

Um sistema mnemônico para representar programas é uma linguagem de montagem que representou um passo gigantesco em direção à busca por melhores técnicas de programação. Na verdade as linguagens de montagem foram tão revolucionárias que se tornaram conhecidas como linguagens de segunda geração, sendo a primeira geração as linguagens de máquina propriamente ditas [7].

Entretanto, como as primitivas usadas em uma linguagem de montagem são essencialmente as mesmas que as encontradas na linguagem de máquina, um programa escrito em linguagem de montagem é dependente da máquina, ou seja, as instruções dentro do programa são expressas em termo dos atributos de uma máquina específica. Logo, um programa escrito em uma linguagem de montagem não pode ser facilmente transportado para outro projeto de computador, pois teria de ser reescrito para estar em conformidade com a configuração e com o conjunto de instruções do novo computador [7].

Em resumo, as primitivas elementares nas quais um produto deve ser constituído não são, necessariamente, as primitivas que devem ser usadas durante o projeto do produto. O processo de projeto é mais adequado ao uso de primitivas de alto nível, cada uma delas representando um conceito associado a uma característica principal do produto. Um vez que o projeto tenha sido feito, essas primitivas podem ser traduzidas para conceitos de nível mais baixo, relacionados aos detalhes da implementação [7].

Dessa forma, surgiu uma terceira geração de linguagem de programação que diferia das gerações anteriores no sentido de que suas primitivas eram tanto de alto nível (no sentido de que expressavam instruções em incrementos maiores), quanto eram independentes de máquina (no sentido de que não dependiam das características de uma máquina específica). Como as sentenças em uma linguagem de terceira geração não se referem a atributos de qualquer máquina em particular, elas podem ser compiladas facilmente de uma máquina para outra, ou seja, um programa escrito em uma linguagem de terceira geração poderia, em teoria, ser usado em qualquer máquina simplesmente aplicando o compilador apropriado [7].

Porém, quando um compilador é projetado, características particulares da máquina subjacente são, algumas vezes, refletidas como condições na linguagem sendo traduzida. Por exemplo, as diferentes maneiras pelas quais as máquinas tratam operações de E/S têm historicamente feito que a “mesma” linguagem tenha diferentes características, ou dialetos, em diferentes máquinas, sendo muitas vezes necessário fazer ao menos algumas pequenas

modificações em um programa para movê-lo de uma máquina para outra gerando, assim, um problema de portabilidade [7].

O fato de as linguagens de terceira geração não terem sido capazes de obter uma independência verdadeira da máquina, não reduz o fato de que os aplicativos de software puderam ser transportados de uma máquina para outra com uma relativa facilidade e que o objetivo da independência da máquina acabou sendo apenas uma porta para objetivos mais ambiciosos. Pois o entendimento de que as máquinas poderiam responder a sentenças de alto nível como

*assign TotalCost the value Price + ShippingCharge*

levou os cientistas da computação a sonharem com ambientes de programação que permitiriam que humanos se comunicassem com máquinas em termo de conceitos abstratos, sem a necessidade da tradução desses conceitos em um formato compatível com a máquina. Ademais, os cientistas da computação queriam máquinas que pudessem realizar boa parte do processo de descoberta de algoritmos em vez de apenas sua execução. O resultado tem sido uma evolução constante no campo da linguagem de programação, que segundo [7] desafia uma classificação clara em termos de gerações.

Para Brookshear [7], quaisquer paradigmas identificados possuíram ramificações além do processo de programação. Eles representarão abordagens fundamentalmente diferentes para construir soluções para problemas e, logo, afetarão o processo inteiro de desenvolvimento de software. Nesse sentido, o termo *paradigma de programação* é impróprio. O autor sugere um termo, segundo ele mais realístico, que seria *paradigma de desenvolvimento de software*.

A figura 2.10 enfatiza diversas tendências em software. Inicialmente, há uma tendência maior ao abandono de programas personalizados desenvolvidos pelos programadores profissionais de uma organização, passando ao uso de pacotes de software disponíveis adquiridos por usuários finais com fornecedores. Esta tendência aumentou de forma impressionante como o desenvolvimento de pacotes de software relativamente baratos e fáceis de usar, e conjuntos de software para microcomputadores destinados a múltiplas finalidades. A tendência tem se acelerado à medida que os pacotes de software são projetados com capacidades para rede e com características de colaboração que otimizam sua praticidade para usuários finais e grupos de trabalho na internet, intranets e extranets [6].

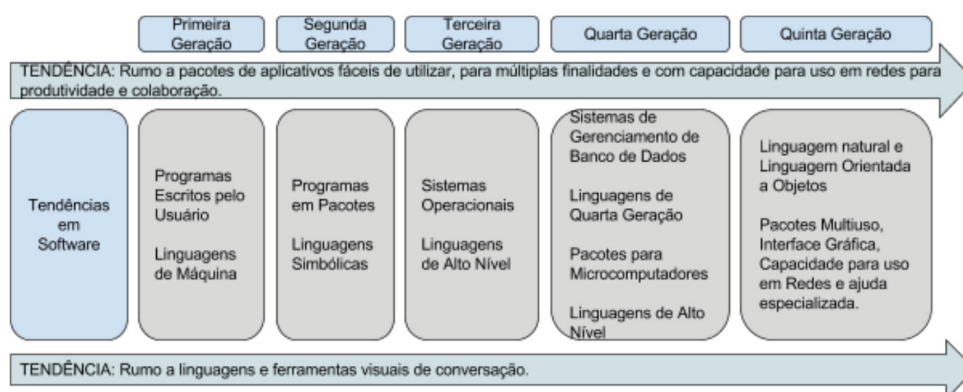


Figura 2.10: Tendências em software. Fonte:[6]

Identifica-se ainda uma tendência definida no sentido do abandono de linguagens de programação técnicas e específicas à máquina, utilizando código a base binária ou simbólico, ou linguagens procedurais, que utilizam formulações breves e expressões matemáticas para especificar a sequência de instruções que um computador deve executar. Em seu lugar, a

tendência passa a ser usar uma interface gráfica visual para programação que se aproxima mais da conversação humana. Esta tendência se acelerou com a criação das linguagens de quarta geração (4GLS), não procedurais e fáceis de utilizar. A tendência continua a crescer à medida que avanços na tecnologia orientada a objetos, gráficos e inteligência artificial produzem linguagem natural e interfaces gráficas com o usuário, facilitando tanto a utilização de ferramentas como de pacotes de software de programação [6].

### *Software Livre e o movimento Open Source - tipos de licenças*

Com o surgimento dos primeiros computadores vendidos comercialmente, a partir da década de 1950, foram criados também os primeiros programas que iriam ser executados neles. Muitas vezes ocorria uma venda casada entre hardware e software, pois os programas eram fortemente acoplados à arquitetura das máquinas em que eram executados. Nessa época, o foco das empresas era na venda do hardware, e não eram colocadas muitas restrições no uso que as pessoas fariam do software. Elas podiam adaptá-lo como quisessem, de forma a fazer melhor uso do hardware que tinham disponível, sem sofrer repreensões [60].

Na década de 1970 a situação começou a se modificar. Entre 1968 e 1969 a IBM anunciou o *unbundling* de seus produtos, separando o software do hardware, e já haviam empresas que focavam no desenvolvimento de software. Algumas dessas empresas, como a *Microsoft*, não estavam satisfeitas com a forma como seus programas eram muitas vezes redistribuídos sem que a empresa recebesse royalties pelas cópias. Assim, em 3 de fevereiro de 1976, *Bill Gates* escreveu a *Open Letter to Hobbyists*<sup>1</sup>, que foi publicada na *newsletter* do *Homebrew Computer Club*. Nessa carta, *Bill Gates* afirma que o total de royalties recebidos pelo *Altair BASIC* era equivalente a apenas dois dólares por hora gasta em seu desenvolvimento e documentação. Ele ainda alega que a prática de compartilhamento de software não é justa e afirma que tal prática evita que software bem feito seja escrito. Assim, nessa época, começou uma mudança de postura na indústria, que passou a proibir que o software fosse copiado ou modificado. Surgiu então o que conhecemos por software fechado, caracterizado pelas restrições que são feitas à forma como ele será utilizado.

Como resposta a essa nova situação, surgiram iniciativas voltadas para retomar a liberdade de melhorar e compartilhar o software. Discutiremos a seguir duas delas: o Projeto GNU, combinado com a filosofia do Software Livre, e o movimento *Open Source*. Nesse movimento de retomada de liberdade e compartilhamento do software, o detentor dos direitos patrimoniais sobre um software, quando decide torná-lo livre, deve escolher os termos em que seu trabalho será distribuído, ou seja, os direitos que ele estará transferindo para as outras pessoas e quais as condições que serão aplicadas. O documento que formaliza esse ato é a licença, que normalmente é distribuída junto com o código fonte.

Há inúmeras possibilidades para redigir o texto de uma licença de software livre, mas a prática mais comum e recomendada é reaproveitar alguma das licenças já consolidadas na comunidade. Dessa forma, reduz-se a proliferação de licenças, que deve ser evitada pois gera trabalho adicional para os usuários, uma vez que torna-se necessário para eles estudar os termos de cada nova licença presente no software que irão utilizar. No portal *SourceForge*<sup>2</sup>, que hospeda um grande número de projetos de software livre, ao criar um novo projeto são apresentadas oito opções de licenças, que são as mais utilizadas naquele repositório, além da opção de domínio público:

- GNU *General Public License* (GPL)

<sup>1</sup>[https://pt.wikipedia.org/wiki/Carta\\_Aberta\\_aos\\_Hobbystas](https://pt.wikipedia.org/wiki/Carta_Aberta_aos_Hobbystas) acessado em 02/1/2016.

<sup>2</sup><https://sourceforge.net> acessado em 02/11/2016.

- *GNU Library or Lesser General Public License (LGPL)*
- *BSD License*
- *MIT License*
- *Apache License V2.0*
- *Artistic License*
- *Mozilla Public License 1.1 (MPL 1.1)*
- *Academic Free License (AFL)*

A *Open Source Initiative* (OSI) mantém uma lista de licenças aprovadas de acordo com a definição de código aberto, organizadas em diversas categorias, além de uma lista de licenças atualmente buscando aprovação. A OSI também está trabalhando com um comitê de proliferação de licenças, buscando separar e dar destaque a algumas licenças que passarão a ser recomendadas, em oposição às meramente aprovadas [61].

Considerando que uma das questões mais importantes é a compatibilidade entre licenças [62], vamos separar as licenças descritas aqui em três categorias, de acordo com a presença de termos que impõem restrições de licenciamento na redistribuição do trabalho ou criação de trabalhos derivados. Em relação a essa característica, as licenças são consideradas permissivas ou recíprocas, sendo que entre as recíprocas devemos ainda considerar que algumas forcem que seja mantida a mesma licença em mais casos do que outras e, assim, as dividimos entre recíprocas parciais, que também recebem a denominação de *copyleft* fraco, e recíprocas totais.

Tabela 2.1: Categoria de licenças

Licenças	Significado
Licenças Permissivas	As licenças permissivas, também chamadas de licenças acadêmicas por alguns autores, como Rosen [63] e Laurent [63], em referência às origens das licenças BSD ( <i>University of California, Berkeley</i> ) e MIT ( <i>Massachusetts Institute of Technology</i> ), impõe poucas restrições às pessoas que obtém o produto. Licenças permissivas são uma ótima opção para projetos cujo objetivo é atingir o maior número de pessoas, não importando se na forma de software livre ou de software fechado. Temos vários exemplos desse modelo no BSD Unix, que continha o software de TCP/IP que hoje é usado na maior parte das implementações desse protocolo, incluindo a da <i>Microsoft</i> [63].
Licenças Recíprocas Totais	Determinam que qualquer trabalho derivado precisa ser distribuído sob os mesmos termos da licença original. Isso também é chamado de <i>copyleft</i> , um termo criado pela <i>Free Software Foundation</i> ( <a href="http://www.gnu.org/copyleft">www.gnu.org/copyleft</a> )

*Continua na próxima página*

Tabela 2.1 – *Continuação*

Licenças	Significado
Licenças Recíprocas Parciais	Também chamadas de <i>copyleft</i> fraco, determinam que modificações do trabalho coberto por elas devem ser disponibilizadas sob a mesma licença. Considera-se que essas licenças são as que melhor equilibram dois fatores importantes do modelo de software livre: atração de interesse para a comunidade e força e longevidade do código fonte disponível. Ao mesmo tempo que essas licenças permitem que os desenvolvedores utilizem o trabalho para criar software que será licenciado como preferirem, modificações e melhorias feitas ao próprio trabalho são obrigatoriamente disponibilizadas à comunidade. A <i>Free Software Foundation</i> recomenda o uso desse tipo de licença apenas em casos específicos. Seu argumento é a necessidade de fortalecer o software livre em detrimento do software fechado. Assim, quando as funcionalidades de uma biblioteca não estão facilmente disponíveis para uso em software fechado, seria melhor mantê-las dessa forma, utilizando uma licença recíproca total.

Tabela 2.2: Tipos de licenças na categoria das licenças permissivas

Licenças	Descrição	Fonte
BSD	Foi a primeira licença de software livre escrita e até hoje é uma das mais usadas. Criada originalmente pela Universidade da Califórnia em <i>Berkeley</i> para seu sistema operacional derivado do UNIX chamado <i>Berkeley Software Distribution</i> . Os principais motivos que levaram a licença BSD a ser tão difundida são a simplicidade de seu texto e o fato dela ter sido inicialmente adotada por um projeto amplamente disseminado, o que criou um ciclo virtuoso em que mais comunidades a adotaram, tornando-a ainda mais reconhecida.	<a href="http://www.opensource.org/licenses/bsd-license.php">www.opensource.org/licenses/bsd-license.php</a>

*Continua na próxima página*



Tabela 2.2 – Continuação

Licenças	Descrição	Fonte
MIT/X11	Criada pelo <i>Massachusetts Institute of Technology</i> , é também conhecida como Licença X11 ou X, por ter sido redigida originalmente para o <i>X Window System</i> , desenvolvido no MIT em 1987. Essa também é uma licença permissiva e é considerada equivalente à BSD Simplificada sem a cláusula de endosso. Porém, seu texto é bem mais explícito ao tratar dos direitos que estão sendo transferidos, afirmando que qualquer pessoa que obtém uma cópia do software e seus arquivos de documentação associados pode lidar com eles sem restrição, incluindo sem limitação os direitos a usar, copiar, modificar, mesclar, publicar, distribuir, sublicenciar e/ou vender cópias do software. As condições impostas para tanto são apenas manter o aviso de copyright e uma cópia da licença em todas as cópias ou porções substanciais do software.	<a href="http://www.opensource.org/licenses/mit-license.php">www.opensource.org/licenses/mit-license.php</a>
Apache	A <i>Apache</i> também é uma licença permissiva e, na versão 1.1, seu texto era bastante similar ao da BSD, porém dando ênfase à proteção da marca <i>Apache</i> .	<a href="http://www.apache.org/licenses/LICENSE-2.0.html">www.apache.org/licenses/LICENSE-2.0.html</a>

Tabela 2.3: Tipos de licenças na categoria das licenças recíprocas totais

Licenças	Descrição	Fonte
GPL, GPL 2.0	A licença GPL pode ser copiada, distribuída e aplicada a qualquer software cujo detentor dos direitos autorais assim desejar. Porém, diferentemente de outras licenças, como a BSD, o texto da GPL não pode ser alterado sem autorização, ou seja, não é permitido que seja feita uma licença derivada dela. No final da licença, há uma explicação sobre como aplicá-la a um trabalho. A GPL 2.0 inclui um preâmbulo que explica os princípios que baseiam a licença e seus principais objetivos. Apesar desse preâmbulo ser bastante citado nas discussões, seu valor jurídico é menor que o restante da licença, pois, por não fazer parte dos termos e condições, suas palavras não precisam ser obedecidas por quem obtém a licença do software. Seu objetivo é apenas melhorar o entendimento da GPL em seu contexto [64], explicando o que é software livre e a importância do <i>copyleft</i> . Porém, ele pode ser utilizado para esclarecer a vontade do licenciante no caso de uma disputa ir a tribunal.	<a href="http://www.gnu.org/licenses/old-licenses/gpl-1.0.html">www.gnu.org/licenses/old-licenses/gpl-1.0.html</a>

Continua na próxima página

Tabela 2.3 – Continuação

Licenças	Descrição	Fonte
GPL v3	Lançada em 29 de junho de 2007, após um longo período de discussão e revisão pública, foi criada para evitar algumas situações consideradas indesejáveis pela <i>Free Software Foundation</i> . Além disso, algumas partes foram reescritas de forma a adaptar a licença a novas formas de compartilhamento de programa e a deixá-la mais adequada para legislações em que os termos originais poderiam ser interpretados de forma diferente da esperada pela <i>Free Software Foundation</i> . Também foram realizadas alterações para facilitar a compatibilidade com outras licenças, em particular a Apache 2.0.	<a href="http://www.gnu.org/licenses/gpl-3.0-standalone.html">www.gnu.org/licenses/gpl-3.0-standalone.html</a>
AGPL	A Affero Inc. ( <a href="http://www.affero.org">www.affero.org</a> ) é uma empresa que se define com a missão de “trazer a cultura de patrocínio para a Internet”. Ela provê um serviço de hospedagem de páginas pessoais para autores de diversos tipos e integra um sistema de pagamento seguro para que pessoas possam fazer doações. A Affero apóia o desenvolvimento de software livre e, em março de 2002, criou a primeira versão da <i>Affero General Public License</i> , ou AGPL. A AGPL é uma adaptação da GPL, autorizada pela <i>Free Software Foundation</i> , que inclui um termo sobre uso de um software através de uma rede.	<a href="http://www.affero.org/oagpl.html">www.affero.org/oagpl.html</a>
<i>European Union Public License</i>	A União Européia lançou, em janeiro de 2007, a <i>European Union Public License</i> 1.0 (EUPL), baseada na <i>Open Software License</i> (OSL) versões 2.1 e 3.0.	<a href="https://joinup.ec.europa.eu/page/eupl-guidelines-faq-infographics">https://joinup.ec.europa.eu/page/eupl-guidelines-faq-infographics</a>

Tabela 2.4: Tipos de licenças na categoria das licenças recíprocas parciais

Licenças	Descrição	Fonte
LGPL	A GNU <i>Lesser General Public License</i> , ou LGPL originalmente denominada GNU <i>Library General Public License</i> , foi escrita em 1991 pela <i>Free Software Foundation</i> . Assim como a GPL e a AGPL, passou por grandes modificações no final de 2007 para adequar-se à versão 3 das licenças.	<a href="http://www.fsf.org/licenses/lgpl.html">www.fsf.org/licenses/lgpl.html</a>

Continua na próxima página



Tabela 2.4 – Continuação

Licenças	Descrição	Fonte
Mozilla	A licença <i>Mozilla</i> é considerada bem escrita e serviu como modelo para muitas das licenças de software livre comerciais que a seguiram [64]. Ela une características de licenças recíprocas e de licenças permissivas, e, portanto, também é categorizada como uma licença recíproca parcial. Na licença <i>Mozilla</i> , a delimitação é bastante clara: o código coberto pela licença deve ser redistribuído pelos termos da licença <i>Mozilla</i> , porém esse código também pode ser utilizado em trabalhos ampliados, que podem estar sob outra licença.	<a href="https://www.mozilla.org/en-US/MPL/">https://www.mozilla.org/en-US/MPL/</a>
Eclipse	Assim como a <i>Netscape</i> , a IBM é uma organização para a qual o software livre tornou-se peça chave do posicionamento estratégico, e neste sentido usou seus recursos para dar um tratamento diferenciado aos aspectos legais envolvendo esse tipo de software. A <i>Eclipse Public License</i> (EPL) é a terceira geração de licenças de software livre originadas na empresa.	<a href="https://www.eclipse.org/legal/epl-v10.html">https://www.eclipse.org/legal/epl-v10.html</a>

A complexidade de licenças de um projeto grande pode ser bastante alta. Numa distribuição *Linux* (por exemplo, a *Debian Etch*) há cerca de 20 mil pacotes e são encontradas neles mais de 300 mil declarações de licença [65], que vão desde simples referências, tais como “este software está licenciado sob a GPLv3” até textos completos da licença e ainda alguns casos de negação, como “este software não está sob a GPL”.

Um desafio normalmente encontrado por quem coordena o desenvolvimento em um projeto que utiliza diversos componentes é ter visibilidade e controle sobre o que está sendo utilizado pela equipe de programadores, principalmente no caso de projetos de software livre, em que os processos tendem a ser menos rígidos devido à forma como a comunidade se organiza. Algumas ferramentas foram criadas para facilitar a tarefa de analisar as licenças envolvidas em um projeto ou determinar quais seriam as opções de escolha de licença em um determinado caso. Com isso, a ideia é minimizar erros numa verificação manual das licenças, além de poupar tempo e dinheiro que seriam gastos no caso de um litígio decorrente de uma cláusula de licença que não foi seguida corretamente. Segue abaixo uma lista com nome e site de algumas ferramentas, que podem ser utilizadas isoladamente ou em conjunto [65].

Tabela 2.5: Tipos de ferramentas para analisar as licenças existentes em um software

Ferramenta	Fonte
FOSSology	<a href="http://www.fossology.org">www.fossology.org</a>
Licensator	<a href="http://licensator.appspot.com">licensator.appspot.com</a>
OSOR.EU License Wizard	<a href="http://www.osor.eu/legal-questions-1/licence-wizard">www.osor.eu/legal-questions-1/licence-wizard</a>
Carneades	<a href="http://carneades.berlios.de">carneades.berlios.de</a>
Open Source License Checker	<a href="http://forge.ow2.org/projects/oslc3">forge.ow2.org/projects/oslc3</a>
BlackDuck	<a href="http://www.blackducksoftware.com">www.blackducksoftware.com</a>

*Continua na próxima página*

Tabela 2.5 – Continuação

Licenças	Fonte
Palamida	(www.palamida.com)
Microformatos	microformats.org.
Open Compliance Program	git.linuxfoundation.org

### Qualidade de software

Para [24], a qualidade de software pode ser definida como uma gestão de qualidade efetiva aplicada de modo a criar um produto útil que forneça valor mensurável para aqueles que produzem e para aqueles que o utilizam<sup>3</sup>.

A qualidade de um "produto de software" pode ser definida pela totalidade de características que lhe confere a capacidade de satisfazer as necessidades implícitas de seus usuários, podendo ser avaliada em três diferentes perspectivas: qualidade externa, qualidade interna e qualidade em uso [66].

A qualidade interna é a perspectiva da qualidade visível aos desenvolvedores e é avaliada nos produtos intermediários do software (diagramas, códigos de programas, documentação etc). As métricas internas devem ser escolhidas pelo desenvolvedor do software. Como a qualidade de um produto depende fortemente da qualidade do seu processo de desenvolvimento, a qualidade interna irá refletir na qualidade externa do software [67].

A qualidade externa é a perspectiva da qualidade visível aos usuários do sistema. Os aspectos que definem a qualidade externa devem ser avaliados em tempo de execução do software e, por isso, a avaliação englobará um sistema em execução que envolve software e hardware. Desta forma, os valores das métricas dependem de fatores além do software, já que se avalia o software como parte de um sistema em operação [67].

Já a qualidade em uso é o efeito combinado das características de qualidade internas e externas de um software em uma situação particular de uso. A qualidade em uso não é medida por meio das propriedades do software e sim pela capacidade do software de atingir metas específicas com efetividade, produtividade, segurança e satisfação em um contexto de uso especificado [67].

Uma das grandes dificuldades em se medir a qualidade em sistemas computacionais, se encontra na diferença que existe entre produtos de software e produtos manufaturados. De acordo com [68], são listadas algumas características inerentes a essência do software e diferenças relacionadas a um produto manufaturado.

- complexidade: normalmente um produto de software possui muitas linhas de código, muitas regras de negócios e diversos desenvolvedores envolvidos com diferentes idéias;
- invisibilidade: o software é invisível para o usuário, o que se vê são as consequências da execução do sistema. Os próprios desenvolvedores necessitam utilizar modelos para representar o sistema em questão;
- conformidade: o software é a interface entre diversas entidades do meio no qual será utilizado: equipamentos, outros produtos de software, usuários e cultura organizacional;
- produção: para software não existe produção em série, pois dependendo do cliente o sistema necessita de aplicações específicas;

<sup>3</sup>Essa definição foi adaptada de [Bes04] e substitui uma visão mais voltada para a manufatura apresentada em edições anteriores deste livro. [Essa nota de rodapé está no livro do Pressman Ed. 8º, pg. 414].

- durabilidade: os softwares são duráveis, as falhas presentes nos sistemas computacionais acontecem devido a erros de implementação;
- validade: o software não é sensível a problemas ambientais e nem sofre qualquer tipo de defeito devido ao efeito acumulativo de seu uso;
- custo do software: cópias do software podem ser reproduzidas em segundos e distribuídas a vários clientes, com o custo unitário do projeto e do desenvolvimento.

Quando analisamos a qualidade de um produto físico, é mais fácil determinar padrões e métricas de qualidade do que em um sistema computacional, pois quando se trata de um software, existe uma dificuldade maior em se saber o que medir e o que comparar para chegar a uma definição do que é qualidade [69].

Em relação ao software, a qualidade de seus produtos pode ser definida como um conjunto de características que devem ser alcançadas para que o produto possa atender as necessidades implícitas e explícitas dos clientes. Para tratar das necessidades dos clientes e aspectos de qualidade do produto de software, existem normas internacionais que auxiliam e identificam aspectos relevantes sobre a mensuração desta qualidade. Como exemplos, podemos citar as normas ISO/IEC 9126 e a ISO/IEC 14598 [70], que respectivamente tratam da qualidade destes produtos trazendo diversos requisitos, servindo como um modelo de qualidade; e abordando uma metodologia própria de avaliação.

Por não ser trivial a avaliação de qualidade em um sistema computacional, existem diversos “Métodos, modelos e normas de avaliação de qualidade”, a exemplo:

Tabela 2.6: Exemplos de método, modelo e/ou norma. Fonte: o autor com base na ISO/IEC 9126, 14598, 25000, Paradigma GQM, Modelo Rocha e Método Mede Pros

<b>Método, modelo e/ou norma</b>	<b>Descrição</b>
Norma ISO/IEC 9126	é uma das normas mais antigas na área de qualidade de software. Ela fornece seis características divididas em sub-características, tendo o objetivo de estabelecer um modelo de qualidade com os seguintes componentes: processo de desenvolvimento; produto; qualidade em uso.
ISO/IEC 14598	é um guia para avaliação da qualidade em produtos de softwares. Sua principal finalidade é a definição de um processo de avaliação baseado na utilização da norma ISO 9126. O processo de avaliação proposto pode ser utilizado tanto para avaliar produtos já existentes ou em desenvolvimento, independentemente do domínio ao qual pertençam. A norma ISO 14598 é constituída por seis documentos distintos, sendo: ISP IEC 14598-1 (visão Geral), 14598-2 (Planejamento e gerenciamento), 14598-3 (Guia para Desenvolvedores), 14598-4 (Guia para Aquisição), 14598-5 (Guia para Avaliação) e 14598-6 (Módulos de Avaliação).

*Continua na próxima página*

Tabela 2.6 – *Continuação*

<b>Método, modelo e/ou norma</b>	<b>Descrição</b>
ISO/IEC 25000 (SQuaRE)	pode ser considerada como uma das normas mais relevantes no sentido de medir e avaliar a qualidade em produtos de software. Esta norma é uma evolução das normas ISO/IEC 9126 e da ISO/IEC 14598, onde também é conhecida como SQuaRE ( <i>Software Quality Requirements and Evaluation</i> ). Para suprir uma falta de clareza e uma difícil compreensão pelos usuários em relação à norma ISO/IEC 9126 e a ISO/IEC 14598, em 1998, o grupo de trabalho WG6 do Subcomitê de Sistemas e Software (SC7) da ISO/IEC, responsável pela elaboração de normas internacionais que tratam da especificação, medição e avaliação da qualidade de produtos de software, propôs a criação de um guia reformulado que objetivava melhorar a compreensão dos usuários.
Paradigma GQM	o modelo GQM ( <i>Goal, Question, Metric</i> ), é uma linha de pesquisa que vem a acrescentar benefícios no gerenciamento de qualidade em grandes organizações. O GQM propõe um planejamento estratégico com interação de produtos e processos, para que se obtenha alta aceitação do cliente e baixa disfuncionalidade organizacional através de mercados em diferentes países.
Método Mede-Pros	O MEDE-Pros [1996], pode ser considerado um método de avaliação da qualidade de software, permitindo efetuar a avaliação da qualidade baseado na ISO/IEC 9126. Basicamente esse método é composto por uma lista de verificação, manual do avaliador e um modelo de relatório de avaliação. De acordo com [69], cada parte do modelo pode ser explicado da seguinte forma: lista de verificação; manual do avaliador e modelo do relatório.

Existem ainda diversos guias de referência que podem ser usados para assegurar a qualidade de software, tais como o *Project Management Body of Knowledge* (PMBok) e o *Capability Maturity Model* [66], mas que servem especificamente para orientar desenvolvedores de software na garantia da qualidade interna e na gestão do processo desoftware. Além desses modelos, a avaliação da qualidade interna pode ser conduzida usando diversos modelos da literatura acadêmica. Dentre esses, incluem-se os modelos de Redmond e Ah-Che [71] e de Pfleeger e Fitzgerald [72], que apoiam o desenvolvimento de métricas e a avaliação da qualidade interna.

Os modelos de qualidade permitem avaliar softwares de acordo com diferentes aspectos e comumente representam as características desejáveis a um software em uma estrutura hierárquica [73]. Algumas normas foram criadas para padronizar internacionalmente as características de implementação e avaliação da qualidade de software. A série de normas ISO/IEC 25000:2014 (*Software Quality Requirements and Evaluation*, SQuaRE) [12], dividida em cinco partes, foi desenvolvida para fornecer suporte à definição de requisitos e ao processo de avaliação da qualidade de softwares. As 5 partes divididas são:

- ISO/IEC 2500n - Divisão de gestão de qualidade: os padrões que formam esta divisão definem os modelos, termos e definições mais utilizados pelos demais padrões da série.

- ISO/IEC 2501n - Divisão de modelo de qualidade: os padrões que formam esta divisão apresentam modelos detalhados de qualidade para produtos de software, qualidade em uso e dados.
- ISO/IEC 2502n - Divisão de medição de qualidade: os padrões que formam esta divisão incluem um modelo de referência de medição de qualidade de produto de software, definições matemáticas de medidas de qualidade e orientação prática para sua aplicação.
- ISO/IEC 2503n - Divisão de requisitos de qualidade: o padrão que forma esta divisão ajuda a especificar requisitos de qualidade, os quais podem ser usados no processo de obtenção de requisitos de qualidade para um produto de software a ser desenvolvido ou como entrada para um processo de avaliação.
- ISO/IEC 2504n - Divisão de avaliação de qualidade: os padrões que formam esta divisão fornecem requisitos, recomendações e diretrizes para avaliação de produtos de software.

O modelo de qualidade apresentado na ISO/IEC 25010:2014 [12] aborda duas dimensões da qualidade de software. A primeira delas é a qualidade em uso, que especifica características de qualidade relacionadas à interação humana com o sistema. A qualidade em uso se refere à capacidade de atender a requisitos para atingir metas específicas com produtividade, efetividade, segurança e satisfação do usuário, em um contexto de uso especificado [12].

A segunda dimensão proposta pela ISO/IEC 25010 é a qualidade do produto, que define um conjunto de oito características da qualidade relacionadas à atributos internos e externos do software, sendo [12]:

- a adequação funcional é a capacidade do software de prover funções que atendam às necessidades implícitas e explícitas quando usado em condições especificadas,
- a eficiência de desempenho se refere ao desempenho em relação à quantidade de recursos utilizados;
- a confiança é a capacidade de manter um nível de desempenho especificado em um determinado período de tempo;
- a usabilidade é a capacidade de ser usado para atingir metas específicas com efetividade, produtividade e satisfação do usuário.
- a manutenibilidade é a capacidade de ser modificado visando à melhoria, correção ou adaptação a mudanças no ambiente ou nos requisitos;
- a portabilidade é a capacidade de ser transferido de um hardware, sistema operacional ou ambiente de uso para outro;
- a compatibilidade é a capacidade de trocar informações com outros softwares e desempenhar as funções que lhe forem requeridas enquanto compartilha recursos de hardware e software;
- a segurança diz respeito à capacidade do software de proteger informações para que pessoas ou outros softwares tenham o nível de acesso apropriado aos seus níveis de permissão.

Conforme descreve o Quadro 2.7, o modelo de qualidade apresentado na ISO/IEC 25010 [12] desdobra as características da qualidade em um conjunto de subcaracterísticas relacionadas.

Tabela 2.7: Características e subcaracterísticas da qualidade de software.Fonte: [12]

CARACTERÍSTICA	SUBCARACTERÍSTICA	DEFINIÇÃO DA SUBCARACTERÍSTICA
Adequação Funcional	Integridade Funcional	Capacidade de atender às tarefas e aos objetivos específicos do usuário a que foi destinado.
	Correção Funcional	Capacidade de fornecer resultados corretos e com precisão.
	Adequação Funcional	Capacidade de realizar tarefas e certos objetivos de maneira fácil.
Eficiência de Tempo	Comportamento em relação ao tempo	Capacidade de fornecer tempos de resposta e processamento apropriados quando o software executa suas funções.
	Utilização de Recursos	Grau com que os tipos e as quantidades de recursos usados atende aos requisitos.
	Capacidade	Grau com que a capacidade máxima de parâmetros do software atende aos requisitos.
Usabilidade	Reconhecibilidade	Grau com que o usuário é capaz de reconhecer se o produto é adequado às suas necessidades.
	Apreensibilidade	Capacidade que o software possui de ser usado para alcançar objetivos específicos de aprendizagem, de forma eficiente, eficaz e sem riscos, garantindo que o usuário se sinta satisfeito no contexto em questão.
	Operacionalidade	Capacidade do software de permitir ao usuário operá-lo e controlá-lo de forma fácil.
	Proteção de erro	Capacidade de proteger os usuários de cometer falhas.
	Estética da interface	Capacidade de possuir uma interface que seja satisfatória ao usuário.
	Acessibilidade	Capacidade de ser usado por usuários com diferentes características e habilidades para alcançar objetivos especificados.
Compatibilidade	Coexistência	Capacidade de compartilhar recursos com outro software sem causar impactos sobre qualquer outro produto.
	Interoperabilidade	Capacidade de interagir com um ou mais sistemas especificados.
Confiança	Maturidade	Capacidade de atender às necessidades de confiabilidade quando operado em condições normais.
	Disponibilidade	Capacidade de ser operacional e acessível quando requerido para uso

*Continua na próxima pagina*



Tabela 2.7 – Continuação

CARACTERÍSTICA	SUBCARACTERÍSTICA	DEFINIÇÃO DA SUBCARACTERÍSTICA
	Tolerância a Falhas	Capacidade de garantir um nível de desempenho especificado em caso de ocorrência de falhas de software ou hardware.
	Capacidade de recuperação	Capacidade de restabelecer seu nível de desempenho especificado e recuperar os dados diretamente afetados no caso de uma falha.
Segurança	Confidencialidade	Capacidade de permitir acesso de dados somente a usuários autorizados.
	Integridade	Capacidade de bloquear acesso e modificações de usuários não autorizados.
	Não repúdio	Capacidade de comprovar ações, eventos, alterações e envio de informações para que não possam ser repudiados futuramente.
	Responsabilidade	Capacidade de rastrear as ações de entidades específicas.
	Autenticidade	Capacidade de comprovar a identidade de um sujeito ou recurso caso seja requerido.
Manutenibilidade	Modularidade	Capacidade de alterar elementos do software com impacto mínimo.
	Reutilização	Capacidade que os componentes do software possuem de serem utilizados por outros sistemas existentes ou em construção.
	Analísabilidade	Capacidade de avaliar o impacto de uma mudança em uma ou mais partes de um sistema, diagnosticar partes do sistema em que podem haver falhas e identificar componentes a serem modificados.
	Modificabilidade	Capacidade de permitir que uma modificação seja implementada sem causar defeitos no produto existente.
	Testabilidade	Capacidade de estabelecer critérios de teste para um sistema que foi modificado e de determinar se esses critérios foram cumpridos de forma eficaz e eficiente.
Portabilidade	Adaptabilidade	Capacidade de ser adaptado para ambientes de operação especificados, sem a necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo software considerado.
	Instalabilidade	Capacidade de ser corretamente instalado e / ou desinstalado em um ambiente especificado.

*Continua na próxima pagina*



Tabela 2.7 – Continuação

CARACTERÍSTICA	SUBCARACTERÍSTICA	DEFINIÇÃO DA SUBCARACTERÍSTICA
	Substituibilidade	Capacidade de substituir outro software no mesmo ambiente e para o mesmo fim.

## 2.6 Aprisionamento tecnológico (*lock-in*)

No momento em que as organizações tomam decisões pela mudança ou das suas estratégias, ou dos seus processos, ou das tecnologias aplicadas, elas têm que levar em conta os custos envolvidos nessa troca. Estes custos de troca podem conduzir a uma decisão de manutenção, anulando o processo de mudança, ou seja, levando a um aprisionamento à situação atual [9].

Ao longo deste levantamento bibliográfico, constatou-se a escassez de literatura que aborde o aprisionamento a partir do ponto de vista do comprador que é aprisionado pelo fornecedor e as estratégias que este pode utilizar para minimizar o efeito. Nesse sentido, faz-se necessário analisar a literatura de forma “invertida”, ou seja, tudo que é estratégia de ataque por parte do fornecedor pode ser entendido como alerta e fator motivador de uma estratégia de defesa por parte do comprador. Ao inverter a abordagem, pode-se ampliar a base de conceitos presentes em alguns estudos desenvolvidos sobre aprisionamento.

### 2.6.1 O ciclo do aprisionamento

O aprisionamento é um conceito dinâmico, originando-se nos investimentos feitos e nas necessidades realizadas, em diferentes pontos ao longo do tempo. Os custos de troca podem aumentar ou diminuir com o tempo, mas não permanecem os mesmos, de acordo com [9].

O ciclo de aprisionamento proposto por [9] está ilustrado na Figura 2.11, conforme a seguir:

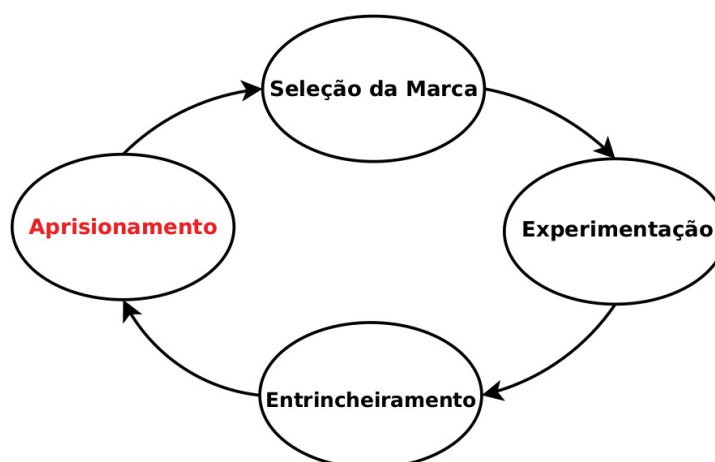


Figura 2.11: Ciclo de aprisionamento. Fonte: [9]

O ciclo do aprisionamento se inicia no momento da seleção de uma marca. Quando se trata da primeira escolha, não existe ainda preferência, mas após a opção pela primeira marca incorre-se no aprisionamento, reduzindo-se a mobilidade para realizar a próxima escolha [9].

Tendo escolhido a marca, a próxima fase é a da experimentação, na qual o usuário estará testando a marca e usufruindo das vantagens e incentivos recebida por ela. Sequencialmente, entra-se na fase do entrincheiramento, que ocorre quando o usuário acostuma-se à nova marca, dando preferência a ela em relação a outras. Quanto mais tempo durar esta fase, maiores serão os custos de troca, levando à fase do aprisionamento, em que os custos de mudança tornam-se altos ao se cogitar a troca de fornecedor, mesmo para setores com custos menores para novos entrantes, como é o caso da área de TI [19].

Ao retornar à escolha da marca, inicia-se um novo ciclo, porém com novos fatores, uma vez que podem existir contratos com os fornecedores formados por compromissos restritivos e por demais onerosos em caso de troca, além de usuários acostumados com a marca utilizada, sendo necessário esforço para eles se habituarem a eventuais alternativas. Os custos de troca tornam-se mais altos, dificultando a migração para novas tecnologias que possam ter surgido no meio tempo.

Dessa forma, para uma empresa diminuir seu aprisionamento frente a determinado sistema de informação, torna-se necessário que ela consiga antecipar vários ciclos futuros ao traçar determinada estratégia, procurando reduzir a dependência de determinada tecnologia [9].

## 2.6.2 Estratégia de aprisionamento para compradores

Os fatores que aprisionam juntamente com os custos de troca podem ser classificados como exógenos ou endógenos. Para cada classificação se faz necessário que as organizações, consumidoras de tecnologia, desenvolvam estratégias distintas para livrar-se dos efeitos inerentes a tais classificações.

A análise de fontes exógenas leva em consideração os fatores do ambiente, ou mercado em geral, que afetam a organização sem que ela possa defender-se adequadamente. Já as fontes endógenas consideram os custos gerados pelas estratégias e pelos processos internos na relação entre fornecedores e compradores [74].

Segundo Shapiro e Varian [9], existem dois elementos fundamentais para a estratégia básica de compradores de tecnologia da informação, que devem ser usados de forma conjunta: (1) negociar antes de ficar aprisionado e (2) manter suas opções abertas, na medida do possível, durante todo o ciclo de aprisionamento.

No momento da negociação, antes de ficar aprisionado, ao tomar decisões de compra e negociar os compromissos contratuais que irão ditar o relacionamento futuro com o fornecedor, as organizações devem analisar todo o ciclo de aprisionamento e não somente os descontos ou ofertas de compra dos sistemas e equipamentos antigos [75].

Com o objetivo de manter sua opção abertas, durante todo o ciclo de aprisionamento, a organização precisa convencer seu fornecedor de que pode mudar a qualquer momento, mesmo que na verdade os custos de troca sejam desestimulantes. Essa estratégia deve ser adotada, na opinião de [9], ao ser ultrapassado a fase do entrincheiramento.

Ainda para manter suas opções abertas e gerir os custos de troca desde o início, a organização deve maximizar as suas opções durante todo o ciclo do aprisionamento. Isto envolve a prospecção, ou mesmo utilização, de um segundo fornecedor e a utilização de sistemas “abertos” e intercambiáveis - deixando claro no planejamento das ações que o termo “aberto”, aqui, significa dispor de acesso irrestrito e sem custos a códigos, documentos, especificações, tecnologias, ou seja, todas as informações necessárias para a continuidade do serviço, sem a necessidade de

manter-se atrelado a um determinado fornecedor e à sua tecnologia proprietária e “fechada” [21] [20].

### 2.6.3 Custo de troca

Hess e Ricart [10] propuseram em seu estudo um conceito de graus de custos de troca que se relaciona com o aprisionamento (*lock-in*), o qual está reproduzido na figura 2.12. Este conceito busca representar a característica dinâmica e contínua dos custos de troca, mostrando que eles podem estar sempre sendo alterados pelas decisões gerenciais.

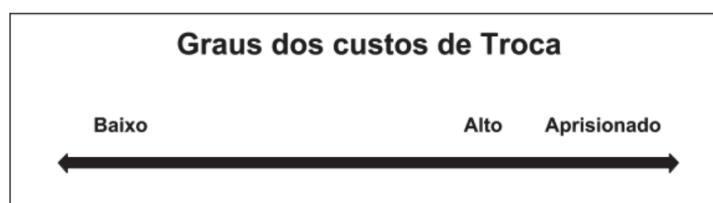


Figura 2.12: Graus dos custos de troca e aprisionamento Fonte:[10]

Os custos de troca tornaram-se a regra e não a exceção nos processos de negócio, pois estes estão cada vez mais baseados em sistemas de informação, que requerem muito investimento, de tempo e de dinheiro, em treinamento e na sua integração [9].

Toda essa especialização nos sistemas atuais torna o processo de troca muito mais custoso, gerando um aprisionamento tecnológico ligado à TI. Ao tomarem decisões pela mudança envolvendo um sistema ou tecnologia, os gestores de tecnologia da informação têm pois que levar em consideração os custos de troca, que geralmente são significativos. Consequentemente, estes custos podem levar a organização a continuar utilizando o sistema ou tecnologia vigente, perpetuando o aprisionamento tecnológico. Na economia da informação, esses custos de troca estão em toda parte e, por meio deles pode-se medir a extensão do aprisionamento de uma organização ao status quo [9], [75], [76].

Klemperer [75], [77] vê os custos de troca como o desejo de compatibilidade entre a nova compra e um investimento já realizado. O autor classifica estes custos em três tipos: custos da transação, custos de aprendizado e custos de troca contratuais ou artificialmente gerados.

Os custos de troca envolvem ambos os tipos de investimentos: os já realizados (passado) e os potenciais (futuro). Esses custos também podem ser arcados tanto por vendedores como por compradores [9]. A classificação proposta por esses autores divide os custos de troca conforme apresentado na Figura 2.13, onde estes são relacionados aos tipos de aprisionamento experimentados pelos consumidores.

Para Hess e Ricart [10], é necessário analisar duas classificações de tipos de custos de troca, como os gerados por investimentos prévios e os relacionados a investimentos potenciais. Ainda é dada relevância adicional a dois conceitos-chave para a análise de custos de troca, acrescentando duas classificações distintas de tipo de custos. São os custos de troca de rede, associados aos investimentos prévios, como aqueles com que os consumidores arcam para fazer parte de uma rede, e os custos de oportunidade que, ao contrário dos tipos citados anteriormente (que resultam de decisões do passado ou de investimentos potenciais), representam os ganhos que se tem com a solução atual e que não permanecerão com a nova opção, embora não se tenha realmente investido diretamente para a obtenção da situação atual.

Tipo de aprisionamento	Custo de troca
Compromissos contratuais	Indenizações compensatórias.
Compra de bens duráveis	O custo de substituição de equipamento tende a cair à medida que o bem durável envelhece (se torna obsoleto).
Treinamento em marca específica	Aprender um novo sistema demanda tempo e incorre em custos, que tendem a aumentar com o tempo.
Informação e banco de dados	O custo da conversão de dados para o novo formato tende a aumentar com o tempo, pois a quantidade de dados aumenta.
Fornecedores especializados	O custo de desenvolvimento de um novo fornecedor tende a ser maior quanto mais difícil for encontrar um novo fornecedor.
Custos de busca	Incluem o aprendizado sobre a qualidade das alternativas, podendo ser incorridos tanto pelo comprador como pelo fornecedor.
Programas de fidelidade	Quaisquer benefícios perdidos do fornecedor, mais a possível necessidade de reconstruir as vantagens do uso cumulativo.

Figura 2.13: Tipo de aprisionamento e custos de troca associados. Fonte: [9]

## 2.7 Considerações do capítulo

Neste capítulo foram apresentados termos e conceitos para compor o entendimento entre a relação da tomada de decisão, por meio da governança corporativa e de TI, e o aprisionamento tecnológico. Optou-se por apresentar 3 dos 5 recursos de TI tendo em vista a abrangência dos temas correlatos e a limitação de testes para compor um conjunto significativo de casos. Dessa forma, foram detalhados os recursos de hardware e software da infraestrutura de TI com o objetivo de identificar os elementos desses recursos que mais contribuem para o fenômeno do aprisionamento (*lock-in*).

### 3 Análise do aprisionamento (*lock-in*) na gestão e infraestrutura de TI

Este capítulo inicia com a demonstração dos diversos momentos onde o aprisionamento pode ser ampliado, com a tomada de decisão no âmbito da governança corporativa e a governança de TI. Observa-se que, quando a tomada de decisão não está amparada na real situação da gestão da infraestrutura de TI, essa decisão poderá ser anulada pela impossibilidade de mudança decorrente da infraestrutura, ou seja, permanecendo aprisionada na situação atual.

Posteriormente neste capítulo é analisado o aprisionamento na gestão de TI ainda como processo de decisão, para em seguida localizar este aprisionamento na infraestrutura de TI, mais especificamente nos recursos de hardware, de software e recursos humanos.

Em seguida é apresentada uma proposta de guia para o gerenciamento do fenômeno do aprisionamento na infraestrutura. Esse guia aponta os elementos dos recursos de TI que mais contribuem para o aprisionamento, seguido de uma nota que posteriormente será utilizada para encontrar um índice que será detalhado ao final do capítulo. Essa proposta de índice indicará o grau com que a infraestrutura de TI está ou ficará aprisionada, possibilitando aos gestores e TI apoiar com maior precisão a tomada de decisão dos *stakeholders* da organização que poderão planejar ações antecipando o ciclo do aprisionamento ou que reduzam o índice encontrado.

#### 3.1 Governança de TI limitada pelo fenômeno do aprisionamento

Ao observarmos a governança corporativa [25], [16], [26], [14], identificamos princípios norteadores como transparência, equidade, prestação de contas e responsabilidade corporativa que, visando equacionar o “conflito de agência” ou “conflito agente-principal”, buscam assegurar o retorno aos provedores financeiros sobre seus investimentos realizados na organização [14] [27].

A governança garante, então, a dimensão de conformidade à governança empresarial com uma perspectiva histórica da organização, cabendo a governança de negócio, que é a dimensão de desempenho, garantir a perspectiva futura. Ou seja, conforme a Figura 2 2.3 é necessária a preocupação com a estratégia e a visão de futuro para com a organização [1].

A governança corporativa tem o seu desenvolvimento irregular como em qualquer processo evolutivo, necessitando de iniciativas como a Comissão de Cadbury em 1992 e a criação da legislação de *Sarbanes-Oxley* (SOX) de 2002, visando tornar o alto escalão das organizações mais atento e responsável por suas decisões [14], [29], [16].

Nesse contexto da governança, de preocupar-se com o controle e com a transparência, foi impondo-se novos processos para o monitoramento até chegarmos na Governança de TI que está relacionada como um dos componentes da governança corporativa [15], [30], [31], [32], [34].

Podemos identificar a governança de TI como um escopo de conhecimento amplo, que trata de assuntos como medição de desempenho, gestão dos recursos e controle dos riscos, tomadas de decisões, princípios, papéis e gestão do portfólio [36], [3], [29]. Pode-se, ainda, conforme esclarecido no Capítulo 2, observar a governança de TI pelos frameworks e auditoria, pela tomada de decisão e pela relação com a governança corporativa [38].

Ao relacionarmos esses aspectos, identificados após a revisão da literatura, com os conceitos e análises pertinentes ao fenômeno do aprisionamento tecnológico (*lock-in*), observamos estreita relação entre o ambiente da governança de TI com o aprisionamento tecnológico.

No momento que as organizações tomam decisões pela mudança de suas estratégias, dos seus processos, ou das tecnologias aplicadas, que os custos de troca podem levar a uma decisão de manutenção, anulando o processo de mudança, ou seja, levando a um aprisionamento à situação atual [9].

Então é possível constatar a relação entre os três aspectos que envolvem todo o ambiente de governança da TI e o aprisionamento. Os desdobramentos oriundos da governança de TI podem implicar em mudanças que poderão ser anuladas em decorrência do custo de troca.

Conforme a abordagem, as definições sobre governança de TI refletem duas visões: a governança de TI como estrutura de decisões e como conjunto de processos [29].

Ao observar a governança de TI como processo [43], [35], [78], [46], [45], [34], podemos estabelecer uma relação mais clara com o aprisionamento analisando o Modelo de Alinhamento Estratégico - *Strategic Alignment Model* (SAM) Figura 2.2 [2].

Nesse modelo, o ajuste estratégico é composto pela estratégia de negócio com escolhas utilizadas para definir o posicionamento da organização no mercado [2], podendo ser traçado um paralelo com as fontes de aprisionamento exógenas que levam em consideração o ambiente, ou mercado em geral, que afetam a organização sem que ela possa defender-se adequadamente [74].

A integração funcional do SAM, conforme descrição no Capítulo 2, permite à organização alinhar seus processos com vistas a balancear as decisões relacionadas à estratégia de negócio, estratégia de TI, infraestrutura organizacional e de processos e infraestrutura de sistemas de informação. Em relação a essa integração, cabe associar as fontes endógenas de aprisionamento que consideram os custos gerados pelas estratégias e pelo processos internos na relação entre fornecedores e compradores [74].

O alinhamento entre a TI e os negócios, ou o alinhamento estratégico da TI, focaliza o grau em que os investimentos em TI estão em sincronia com os objetivos estratégicos. Em empresas com ativos significativos de TI, esse alinhamento se torna relevante pois visa um desempenho superior nos resultados oriundos da sinergia entre os recursos de TI e os objetivos da empresa [43]. Ou seja, o simples fato de escolher uma marca, poderá implicar no efetivo alinhamento, pois o ciclo de aprisionamento [9] [19] se inicia no momento da seleção de uma marca, reduzindo-se a mobilidade para realizar a próxima escolha.

Quando a governança de TI é observada como estrutura [48], [49], [40], [50], [3], [36], [3], [18], [29], constata-se a busca em atender a gestão e o fornecimento dos serviços de TI que está concentrada sobre a autoridade e o local da tomada de decisão. A definição de quem deve tomá-las e o aparato necessário para que boas decisões sejam tomadas, tornam-se as questões centrais na tomada de decisão [15], [3]. Dessa forma, novamente identificamos a relação intrínseca com o fenômeno do aprisionamento, que trata do momento em que a organização toma a decisão pela mudança e que pode, pelo elevado custo de troca, manter-se aprisionado na situação atual [9].

Ao observarmos as decisões fundamentais em TI conforme Figura 2.3, notamos que se a arquitetura da empresa define dados e infraestruturas como plataforma estável, que dá suporte às aplicações específicas de negócios, principalmente se a missão da empresa não mudar, é



possível afirmar que a observância do ciclo do aprisionamento no momento de definir os dados e a infraestrutura facilitará o suporte às aplicações, uma vez que os custos de troca tornam-se altos, dificultando a migração para novas tecnologias e que a dependência de determinada tecnologia diminuirá se a empresa conseguir antecipar vários ciclos futuros ao traçar determinada estratégia [9], [3], [18].

Ainda nas decisões fundamentais de TI, prover uma infraestrutura de TI adequada significa boa relação custo-benefício, que capacite a empresa a adotar rapidamente novas aplicações de negócio [5]. Novamente a conexão se estabelece com o fenômeno do aprisionamento, na medida em que os custos de troca tornam-se a regra e não a exceção nos processos de negócio, pois estes estão cada vez mais baseados em sistemas de informação, que requerem muitos investimento, de tempo e dinheiro, em treinamento e na sua integração [9].

Por fim, na Figura 2.3, o investimento em TI precisa mais que um retorno mínimo (*Return on Investment* - ROI), a empresa necessita de um valor adicional aos negócios que decorre das mudanças organizacionais que a TI proporciona [5]. Assim, em se tratando de mudanças organizacionais, manter suas opções abertas e gerir os custos de troca desde o início irá maximizar as suas opções durante todo o ciclo do aprisionamento [21] [20].

Tendo contextualizado o aprisionamento tecnológico no ambiente da governança corporativa e na governança de TI, cabe observarmos os seguintes apontamentos:

1º - a decisão tomada considerou uma consolidação prévia dos recursos da infraestrutura de TI?

2º - as opções tecnológicas escolhidas seguem o padrão predominante na infraestrutura? Ou o padrão a ser adotado alterna entre aberto ou fechado tanto para escolha de hardware como de software?

3º - o custo ou a intensidade de esforço a ser exigido da gestão de TI e materializado na respectiva infraestrutura, foi considerado no ato da tomada de decisão?

Observa-se que a decisão a ser tomada deve estar embasada em informações que tenham origem na gestão da TI, mais especificamente no âmbito da infraestrutura, pois esta será o suporte efetivo para concretizar as decisões tomadas. Então, cabe à governança de TI a aproximação e a sincronização com os objetivos da governança corporativa, os quais são usados na criação dos objetivos e métricas necessárias para o adequado gerenciamento da TI.

## 3.2 Aprisionamento na gestão de TI

A partir da análise feita na sessão anterior, podemos afirmar que a TI deverá ser a integração do gerenciamento da TI - que é orientado a serviços - com a governança de TI - que deve alinhar a TI com os objetivos da governança corporativa [53]. Logo, sendo a gestão de TI orientada a serviços, pode-se afirmar que esta gestão será responsável por prover e gerar serviços de manutenção de hardware, software, dados e redes, usando várias competências profissionais, como analistas de sistemas, programadores, arquitetos de informação, consultores e gerentes de projetos, entre outros [52], [4], [53].

Na Figura 2.4, que trata das características e preocupações do gerenciamento operacional e estratégico de [4], observamos que o gerenciamento estratégico contempla a implantação de modificações em aspectos da maneira como a empresa compete e sobrevive no mercado, controlando ações e comportamentos necessários para implantar as mudanças [4]. Dessa forma, se ao implementarmos as mudanças o nível de aprisionamento estiver com um elevado grau de custos de troca [10], será possível constatar uma dificuldade proporcional para a efetiva mudança.

O gerenciamento operacional, conforme Figura 2.4, é o conjunto de processos que cooperam para assegurar a qualidade aos serviços de TI, de acordo com os níveis acordados com

o cliente, envolvendo a escolha de programas de tecnologia, a avaliação de riscos e incertezas, o gerenciamento de projetos e a transferência e implantação da tecnologia. Esse conjunto de processos deve estar balanceado conforme aponta Steele [4], mas também com o objetivo de contemplar estratégias que evitem o aprisionamento, como manter as opções abertas, na medida do possível, durante todo o ciclo de aprisionamento. Manter abertas as opções envolve a prospecção, e mesmo a utilização de um segundo fornecedor e a utilização de sistemas “abertos” e intercambiáveis, significa dispor de acessos irrestritos e sem custo a códigos, documentos, especificações e tecnologias, de tal forma que seja garantido a continuidade do serviço, sem a necessidade de manter-se atrelado a um determinado fornecedor e à tecnologia proprietária e “fechada”[21] [20].

### 3.3 Localizando o aprisionamento tecnológico na infraestrutura de TI

Ao identificarmos os componentes da infraestrutura de TI, segundo a Figura 2.5, observamos um agrupamento dos componentes em camadas física e de gestão [45]. A administração dessas duas camadas, de forma articulada, possibilita que a infraestrutura de TI suporte as demandas internas e externas da organização oriundas das decisões tomadas no âmbito da governança corporativa - e alinhadas pela governança de TI [30], [45]. Nesse processo de suporte às demandas externas e internas, podemos traçar um paralelo com o fatores que aprisionam, também de natureza interna e externa à organização [13], [37].

Uma das características principais da infraestrutura de TI é ser orientada a serviços modulares e configuráveis, justamente por envolver tecnologia, processos e pessoas [49]. O fato de ser orientada a serviços modulares e configuráveis pode ser, também, considerado com uma característica positiva para diminuir o aprisionamento, na medida que possibilita articular soluções que mantenham abertas as opções durante todo o ciclo de aprisionamento a exemplo da prospecção ou utilização de um segundo fornecedor e a utilização de sistemas abertos e intercambiáveis [37] [21] [12].

Dessa forma, sendo a TI orientada a serviços modulares e configuráveis, para o presente trabalho é necessário expandir o detalhamento dessa infraestrutura. Podemos observar que para alguns autores, a infraestrutura de TI pode ser composta por diversas partes, tais como hardware, software, dados e pessoas - constituindo-se de uma parte técnica e outra social que se inter-relacionam de forma a se ajustar e cooperar mutuamente [22].

A Figura 2.6, detalha essa infraestrutura e destaca a relação existente entre seus componentes enfatizando os cinco recursos básicos, sendo eles: pessoas, hardware, software, dados e redes. Com essa expansão no detalhamento da infraestrutura de TI será possível traçarmos um paralelo com a Figura 2.13 de que demonstra a classificação dos tipos de aprisionamento e os custos de troca associados [37], [29].

Na Figura 2.13 os custos de troca envolvem os tipos de investimentos já realizados (passado) e os potenciais (futuro). Ao analisarmos os tipos de aprisionamento [37], observamos: que o aprisionamento por contrato é o mais explícito, onde há um compromisso para compra de um fornecedor específico; já a aquisição de bens duráveis envolve outras questões como equipamentos ou produtos complementares, que aprofunda o aprisionamento pelo pós-venda, assim como pela manutenção e serviços de suporte; para o treinamento em marca específica a contraposição é o treinamento genérico, objetivando a transferência ou reaproveitamento do treinamento para outras marcas; no caso da informação e banco de dados, o custo da conversão de dados para um novo formato tende a aumentar com o tempo na medida em que cresce a

quantidade de dados; para os fornecedores especializados uma das alternativas seria manter a estratégia da duplicidade de fontes que muitas vezes pode coincidir também com os interesses dos fornecedores; para os custos de busca haverá sempre um grau de pura inércia e lealdade do consumidor para com os fornecedores titulares pois os consumidores precisarão de tempo para escolher entre as propostas e correr o risco da mudança; e por fim, com os programas de lealdade, também chamados de “aprisionamento artificiais”, os clientes podem calcular com relativa facilidade os custos com que arcam quando mudam de fornecedores - tanto em termos de perda de recompensas, quanto em termos da redução dos retornos marginais sobre novas transações.

Assim, observa-se na Figura 2.13 que os tipos de aprisionamento se assemelham a alguns componentes da infraestrutura de TI, porém as características do custo de troca estão pautadas em dimensões mais econômicas. Então, existe a necessidade de ajustar a análise entre a Figura 2.6 e 2.13, considerando os propósitos distintos a que pertence cada figura e o objetivo a que se propõe este trabalho, a fim de identificar o aprisionamento na infraestrutura de TI.

Como identificado no Capítulo 2, são cinco os recursos que compõem a infraestrutura de TI, porém o presente trabalho considerou mais oportuno iniciar este estudo com três recursos, sendo eles: recursos de hardware, de software e recursos humanos.

Para efeito de análise do presente trabalho, o recurso de rede não foi considerado em decorrência do alto grau de padronização, exigindo dessa forma um aprofundamento maior. Composto por mídias (cabos como coaxial, fibras óticas, cobre e via rádio ou satélite) e máquinas (modems, multiprocessadores, roteadores, comutadores entre outros), esse recurso está sujeito a escolha de fatores como a topologia da rede [79]. Outro fator que contribui para o elevado grau de padronização foi a adoção da definição da interface e do protocolo *OpenFlow*.

Com *OpenFlow*, os elementos de encaminhamento oferecem uma interface de programação simples que lhes permite estender o acesso e controle da tabela de consulta utilizada pelo hardware para determinar o próximo passo de cada pacote recebido. Dessa forma, o encaminhamento continua sendo eficiente, pois a consulta à tabela de encaminhamento continua sendo tarefa do hardware, mas a decisão sobre como cada pacote deve ser processado pode ser transferida para um nível superior, onde diferentes funcionalidades podem ser implementadas. Essa estrutura permite que a rede seja controlada de forma extensível por meio de aplicações, expressas em software. A esse novo paradigma, deu-se o nome de Redes Definidas por Software, ou *Software Defined Networks* (SDN) [80], [29], [81] [82], [79].

Dessa forma, tendo concentrado a análise em 3 recursos da infraestrutura, se faz necessário identificar em cada um destes recurso os elementos mais significativos ou sensíveis, conforme a característica de cada elemento em cada grupo de recursos a que esse elemento pertence. Para tal, no Capítulo 2 deste trabalho, foram levantados conceitos que tratam das características específicas de hardware e software.

Para o componente hardware foram apontadas algumas configurações possíveis de hardwares, bem como apresentados alguns propósitos de utilização. É necessário ressaltar que não é objeto desse trabalho apresentar de forma exaustiva as configuração existente de hardware, tão pouco todos os propósitos possíveis de aplicação para cada configuração.

Considerando o recurso de software foram apontadas características correspondentes à linguagem de programação, tipos de licenças e qualidade de software, tendo como objetivo levantar elementos que apontem o aumento ou redução do aprisionamento.

No tocante aos recursos humanos, o presente trabalho considera apenas a divisão clássica entre recursos humanos especialistas e típicos. Essa divisão segue o contido na literatura e ao mesmo tempo possibilita extrair resultados mais significativos no momento da aplicação do guia.

### 3.3.1 Aprisionamento nos recursos de hardware

Para o componente hardware entendido como tecnologia para processamento computacional, armazenamento, entrada e saída de dados, podemos mencionar as diversas configurações como grandes mainframes, servidores, computadores de médio porte, computadores pessoais ou PC (Personal Computer) e laptops, assistentes digitais pessoais (PAD's) como *BlackBerry*, *Palm*, *iPhone* ou outros modelos de celulares [22], [29].

Para Brookshear [7], o conjunto de circuitos de computador que controla a manipulação de dados é chamado de Unidade Central de Processamento ou CPU. Conforme a Figura 2.7 uma CPU consiste em três partes, sendo: a unidade lógica e aritmética, a unidade de controle e a unidade de registro. Essa configuração é um esquema básico às diversas configurações de computadores.

Para efeito de exemplificação, pode-se mencionar uma tendência que vem ocorrendo com os supercomputadores. Eles passaram a ser tratados também como “servidores redimensionáveis” na extremidade superior das linhas de produto que começam com as estações de trabalho de mesa. Companhias orientadas para o mercado, como a *Silicon Graphics*, *Hewlett-Packard* e *IBM* possuem um foco muito mais amplo do que apenas o de montar o computador mais rápido e o software do computador de mesa passou a possuir muito mais pontos de sobreposição com o do supercomputador, porque ambos são montados a partir dos mesmos microprocessadores baseados em cache [29].

O poder de computação dos microcomputadores aumentou expressivamente [29] tornando possível a instalação de infraestruturas de hardwares com alta capacidade de processamento por um custo viável para as organizações. Na Figura 2.8, na sua forma mais básica, um cluster é um sistema que compreende dois ou mais computadores ou sistemas (nodos) que trabalham em conjunto para executar aplicações ou realizar outras tarefas, de tal forma que os usuários do agrupamento de máquinas tenham a impressão de que somente um único sistema responde para eles, criando a ilusão de um recurso único (máquina virtual) [31].

Em vez de contar com máquinas personalizadas e redes personalizadas para construir máquinas maciçamente paralelas, a introdução de switches como parte da tecnologia de rede significou que a alta largura da banda de rede e o aumento de escala das redes estavam disponíveis a partir de componentes prontos. Quando combinada com o uso de computadores desktop e disco como dispositivos de computação, poderia ser criada uma infraestrutura de computação muito menos dispendiosa, capaz de resolver problemas muito grandes com uma facilidade para aumentar a escala e para isolar falhas [58].

Ou seja, no caso do hardware alguns elementos se tornaram mais sensíveis quando relacionados com o aprisionamento. Dessa forma, listamos os elementos que devem ser observados com prioridade, porém não exclusividade, quando avaliado o nível do aprisionamento no recurso de hardware:

Tabela 3.1: Elementos dos recursos de hardware considerados críticos ao aprisionamento. Fonte: o autor

Elemento	Descrição
Armazenamento secundário	necessário para expansão da capacidade de armazenamento dos dados
Armazenamento primário	necessário para manter/aumentar o desempenho de acordo com o aumento da demanda

*Continua na próxima página*

Tabela 3.1 – *Continuação*

Elemento	Descrição
Processadores/ microprocessadores	necessário para manter/aumentar o desempenho de acordo com o aumento da demanda
Placa de vídeo (com GPU)	necessário para manter/aumentar o desempenho do processamento gráfico de acordo com o aumento da demanda

Esses elementos podem ser ampliados ou até reduzidos a partir de duas perguntas que o presente trabalho considera relevante, sendo: (1) qual a finalidade ou o propósito do hardware avaliado ou a ser adquirido?; (2) o propósito a que se destina é geral ou específico?

Nas duas perguntas acima o resultado será um imediato redimensionamento na configuração do hardware inclusive considerando os tipos de software que serão executados.

### 3.3.2 Aprisionamento nos recursos de software

Para o recurso software [24], [6], [59], se faz necessário destacar a divisão, sistematização ou categorização do software [59], [24], [6], com o objetivo de entender a relação existente entre ambos e as variações que podem ocorrer dependendo do tipo de computadores e redes utilizados e de quais tarefas específicas serão executadas.

Observa-se uma coincidência entre a Figura 2.9 de O'Brien [6] e a sistematização feita por [24]. Ambos os autores separam duas grandes categorias, sendo uma a de software de sistemas (subdividido em programas para desenvolvimento de sistemas e programas de gerenciamento de sistema) e a outra sendo software aplicativo (subdividido em programas para aplicações específicas e programas de aplicações gerais). Essa separação abstrai a localização “física” do software em algum tipo de hardware e agrupa todos os tipos de programas no conjunto, por assim dizer, dos softwares.

Assim, para o presente trabalho, iremos considerar o software e as suas características (instruções, estrutura de dados e informações descritivas) [24] em um conjunto próprio com o objetivo de abstrair elementos comuns para avaliar o aprisionamento independente de sua sistematização. Dessa forma, será observado o aprisionamento na linguagem de programação, nos tipos de licença e na qualidade de software.

#### *Linguagem de programação - uma evolução contra o aprisionamento*

É possível afirmar que o desenvolvimento de sistemas complexos de software, como sistemas operacionais, software de rede e a vasta quantidade de aplicativos de software disponível hoje, seria impossível se os humanos fossem forçados a escrever programas em linguagens de máquina.

Reverendo brevemente a evolução histórica das linguagens de programação, nota-se o primeiro sinal de aprisionamento eminentemente tecnológico (não se tratando de fornecedor, cadeia de mercado, custos agregados, etc.) que, a partir da criação dos programas chamados montadores foi possível um humano desenvolver um programa de forma mnemônica e, então, convertê-lo em linguagem de máquina por meio do montador. Com isso, as linguagens de montagem foram tão revolucionárias que se tornaram conhecidas como linguagem de segunda geração, sendo a primeira geração as linguagens de máquina propriamente ditas [7].

Entretanto, as instruções dentro do programa são expressas em termos dos atributos de uma máquina específica, logo, um programa escrito em linguagem de montagem não pode ser



facilmente transportado de um projeto de computador, pois teria de ser reescrito para estar em conformidade com a configuração e com o conjunto de instruções do novo computador [7].

Assim, surgiu a terceira geração de linguagem de programação que diferia das gerações anteriores no sentido de que suas primitivas eram tanto de alto nível (no sentido de que expressava instruções em incremento maiores), quanto eram independentes de máquina (no sentido de que não dependiam das características de uma máquina específica). Porém, quando um compilador é projetado, características particulares da máquina subjacente são, algumas vezes, refletidas como condições na linguagem sendo traduzida. Por exemplo, as diferentes maneiras pelas quais as máquinas tratam operações de Entrada e Saída (E/S) têm historicamente feito com que a “mesma” linguagem tenha diferentes características, ou dialetos, em diferentes máquinas, sendo muitas vezes necessário fazer ao menos algumas pequenas modificações em um programa para movê-lo de uma máquina para outra gerando, assim, um problema de portabilidade [7].

Mesmo assim, o fato de as linguagens de terceira geração não terem sido capazes de obter uma independência verdadeira da máquina não reduz o fato de que os aplicativos de software puderam ser transportados de uma máquina para outra com uma relativa facilidade - e que o objetivo da independência da máquina acabou sendo apenas uma porta para objetivos mais ambiciosos [7].

Identifica-se na evolução histórica das linguagens de programação, uma tendência definida no sentido do abandono de linguagens de programação técnicas e específicas à máquina, utilizando código a base binária ou simbólico, ou linguagens procedurais, que utilizam formulações breves e expressões matemáticas para especificar a sequência de instruções que um computador deve executar. Em seu lugar, a tendência está evoluindo para o uso de uma interface gráfica visual para programação que se aproxima mais da conversação humana [7].

Tendo exposto a natureza da linguagem de programação, é necessário observar qual o paradigma e a linguagem utilizada para o desenvolvimento de software no processo de avaliação do aprisionamento. Uma vez que a inabilidade em operar a linguagem utilizada em um determinado software, indica um caminho para o aprisionamento em um primeiro momento ao software para em seguida aprisionar ao hardware na utilização do compilador. Assim, podemos levantar os seguintes questionamentos: em qual paradigma e linguagem o sistema foi desenvolvido? Os usuários especialistas da organização conhecem/conseguem desenvolver, manter e/ou evoluir o sistema com essa linguagem? O repositório no qual o código está é aberto/público ou pago? A documentação está/foi produzida de forma satisfatória?

Na Figura 2.10 podemos observar que há uma tendência maior ao abandono de programas personalizados desenvolvidos pelos programadores profissionais de uma organização, passando ao uso de pacotes de software disponíveis adquiridos por usuários finais com fornecedores. Esta tendência aumentou de forma impressionante com o desenvolvimento de pacotes de software relativamente baratos (em muitos casos gratuitos) e fáceis de usar, e conjuntos de software para microcomputadores destinados a múltiplas finalidades. A tendência tem se acelerado à medida em que os pacotes de software são projetados com capacidades para rede e características de colaboração que otimizam sua praticidade para usuários finais e grupos de trabalho na internet, intranets e extranets [29].

### *Software Livre e o movimento Open Source - tipos de licenças*

Com a tendência de utilizar preferencialmente pacotes de softwares projetados e desenvolvidos por comunidades abertas de colaboração, se faz necessário observar os tipos de licenças oriundos dos movimentos do Software Livre e Open Source. Com o surgimento dos primeiros computadores vendidos comercialmente, a partir da década de 1950, foram criados também os primeiros programas que iriam ser executados neles. Muitas vezes ocorria uma venda



casada entre hardware e software, pois os programas eram fortemente acoplados à arquitetura das máquinas em que eram executados (como visto acima no histórico sobre as linguagens de programação). Nessa época, o foco das empresas era na venda do hardware, e não eram colocadas muitas restrições no uso que as pessoas fariam do software [60].

Como resposta à indústria surgiram iniciativas voltadas para retomar a liberdade de melhorar e compartilhar o software. A exemplo do Projeto GNU, combinado com a filosofia do Software Livre, e o movimento Open Source.

Nesse movimento de retomada de liberdade e compartilhamento do software, o detentor dos direitos patrimoniais sobre um software, quando decide torná-lo livre, deve escolher os termos em que seu trabalho será distribuído, ou seja, os direitos que ele estará transferindo para as outras pessoas e quais as condições em que serão aplicados. O documento que formaliza esse ato é a licença, que normalmente é distribuída junto com o código fonte.

Além de ser mais comum, recomenda-se reaproveitar alguma das licenças já consolidadas na comunidade. Dessa forma, reduz-se a proliferação de licenças, que deve ser evitada pois gera trabalho adicional para os usuários, uma vez que torna-se necessário para eles estudar os termos de cada nova licença presente no software que irão utilizar.

Na Tabela 2.1 é demonstrada as três categorias de licenças de acordo com a presença de termos que impõem restrições de licenciamento na distribuição do trabalho ou criação de trabalhos derivados. Em relação a essa característica, as licenças são consideradas permissivas ou recíprocas, sendo que entre as recíprocas devemos considerar que algumas forcem que seja mantida a mesma licença em mais casos do que outras e, assim, podemos dividir entre recíprocas parciais, que também recebem a denominação de copyleft fraco, e recíprocas totais [61] [62].

No Capítulo 2, nas Tabelas 2.2, 2.3 e 2.4 fica demonstrado o tipo da licença a partir da sua categoria, sua descrição e fonte oficial para aprofundar o detalhamento. Ou seja, na Tabela 2.2 observamos os tipos de licenças enquadradas dentro da categoria de licenças permissivas, na Tabela 2.3 as licenças enquadradas na categoria das licenças recíprocas totais e na Tabela 2.4 as respectivas licenças com características da licenças recíprocas parciais.

Foram necessários, no presente trabalho, apontar as categorias e os respectivo tipos de licenças, pois o gerenciamento dos diferentes tipos de licença dentro de um mesmo projeto de software tornou-se significativamente complexo. Conforme Gobeille [65], um exemplo da complexidade pode ser analisado em uma distribuição Linux, a Debian Etch, que possui cerca de 20 mil pacotes e são encontrados neles mais de 300 mil declarações de licenças.

Na figura 2.5 o presente trabalho levantou os tipos de ferramentas e o respectivo site para demonstrar o esforço já existente em analisar as licenças envolvidas em um projeto ou determinar quais seriam as opções de escolha de licença em um determinado caso.

Cabe então destacar a relevância de uma adequada gestão dessas licenças a fim de evitar o aprisionamento por uma opção errada de licença, ou observar a licença a fim de garantir que o software a ser adquirido esteja em conformidade com os critérios estabelecidos para minimizar o aprisionamento junto ao fornecedor. Nesse aspecto cabe as seguintes perguntas: para o meu projeto de desenvolvimento, qual a categoria e qual é o tipo de licença é mais adequada? Conheço todos os tipos de licenças existentes no software adquirido?

### *Características da qualidade do software que contribuem na redução do aprisionamento*

Outro aspecto a ser incorporado na análise dos elementos pertinentes ao recurso de software que compõe a infraestrutura de TI é a qualidade de software. A qualidade de software pode ser definida como uma gestão de qualidade efetiva aplicada de modo a criar um produto útil que forneça valor mensurável para aqueles que produzem e para aqueles que o utilizam e

pode ser avaliada em três diferentes perspectivas, sendo: qualidade externa, qualidade interna e qualidade em uso [24], [3].

A observação pela qualidade interna apoiará a visão do desenvolvedor e é avaliada avaliada nos produtos intermediários do software como diagramas, códigos de programas e documentação entre outros que irá refletir na qualidade externa do software. Essa qualidade externa é a perspectiva da qualidade visível aos usuários do sistema e que também deverá ser avaliado em tempo de execução do software e, por isso, a avaliação envolverá um sistema em execução que demandará software e hardware. Já a qualidade em uso combinará características da qualidade externa e interna para que o software possa atingir metas específicas como efetividade, produtividade, segurança e satisfação em um contexto de uso especificado.

Por não ser trivial a avaliação de qualidade em um sistema computacional, existem diversos métodos, modelos e normas de avaliação de qualidade. Nesse sentido o presente trabalho apresentou na Tabela 2.6 um levantamento que serviu para repassar os diversos métodos, modelos e normas a fim de identificar elementos que apontassem características a serem observadas a luz do fenômeno do aprisionamento.

Após a verificação na literatura realizada no Capítulo 2, o presente trabalho adotou como fonte de parâmetros para apoiar a identificação do aprisionamento nos recursos de software, a série de normas ISO/IEC 25000:2014 (*Software Quality Requirements and Evaluation, SQuaRE*), dividida em 5 partes, que foi desenvolvida para fornecer suporte à definição de requisitos e ao processo de avaliação da qualidade de softwares.

Como demonstrado na Tabela 2.7 o modelo de qualidade apresentado na ISO/IEC 25010:2014 aborda duas dimensões da qualidade de software. A primeira delas é a qualidade em uso, que especifica características de qualidade relacionadas à interação humana com o sistema. A qualidade em uso se refere à capacidade de atender a requisitos para atingir metas específicas com produtividade, efetividade, segurança e satisfação do usuário, em um contexto de uso especificado [12]. A segunda dimensão proposta pela ISO/IEC 25010 é a qualidade do produto, que define um conjunto de oito características da qualidade relacionadas à atributos internos e externos do software [12].

Ao relacionarmos com a Figura 2.13 que demonstra o tipo e custos de troca associados com as duas dimensões abordadas pela ISO/IEC 25010:2014 [12] demonstradas na tabela 2.7, podemos indicar a característica, subcaracterística e a definição da subcaracterística que irá apoiar na identificação do aprisionamento da infraestrutura de TI, conforme o quadro 3.2 abaixo:

Tabela 3.2: Características da qualidade de software selecionadas para apoiar a identificação do aprisionamento no recurso de software. Fonte: o autor com base na ISO/IEC 25010

CARACTERÍSTICA	SUBCARACTERÍSTICA	DEFINIÇÃO DA SUBCARACTERÍSTICA
Usabilidade	Reconhecibilidade	Grau com que o usuário é capaz de reconhecer se o produto é adequado às suas necessidades.

*Continua na próxima pagina*

Tabela 3.2 – *Continuação*

CARACTERÍSTICA	SUBCARACTERÍSTICA	DEFINIÇÃO DA SUBCARACTERÍSTICA
	Apreensibilidade	Capacidade que o software possui de ser usado para alcançar objetivos específicos de aprendizagem, de forma eficiente, eficaz e sem riscos, garantindo que o usuário se sinta satisfeito no contexto em questão.
	Operacionalidade	Capacidade do software de permitir ao usuário operá-lo e controlá-lo de forma fácil.
	Proteção de erro	Capacidade de proteger os usuários de cometer falhas.
	Estética da interface	Capacidade de possuir uma interface que seja satisfatória ao usuário.
	Acessibilidade	Capacidade de ser usado por usuários com diferentes características e habilidades para alcançar objetivos especificados.
Compatibilidade	Coexistência	Capacidade de compartilhar recursos com outro software sem causar impactos sobre qualquer outro produto.
	Interoperabilidade	Capacidade de interagir com um ou mais sistemas especificados.
Manutenibilidade	Modularidade	Capacidade de alterar elementos do software com impacto mínimo.
	Reutilização	Capacidade que os componentes do software possuem de serem utilizados por outros sistemas existentes ou em construção.
	Analisabilidade	Capacidade de avaliar o impacto de uma mudança em uma ou mais partes de um sistema, diagnosticar partes do sistema em que podem haver falhas e identificar componentes a serem modificados.

*Continua na próxima página*

Tabela 3.2 – *Continuação*

CARACTERÍSTICA	SUBCARACTERÍSTICA	DEFINIÇÃO DA SUBCARACTERÍSTICA
	Modificabilidade	Capacidade de permitir que uma modificação seja implementada sem causar defeitos no produto existente.
	Testabilidade	Capacidade de estabelecer critérios de teste para um sistema que foi modificado e de determinar se esses critérios foram cumpridos de forma eficaz e eficiente.
Portabilidade	Adaptabilidade	Capacidade de ser adaptado para ambientes de operação especificados, sem a necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo software considerado.
	Instalabilidade	Capacidade de ser corretamente instalado e/ou desinstalado em um ambiente especificado.
	Substituibilidade	Capacidade de substituir outro software no mesmo ambiente e para o mesmo fim.

### 3.3.3 Aprisionamento nos recursos humanos

Os recursos humanos, no âmbito da Tecnologia da Informação, é composto basicamente por dois tipos de usuários, sendo usuários finais e especialistas. Os usuários finais (também chamados de usuários ou clientes) são pessoas que utilizam sistemas ou as informações produzidas pelo mesmo. Já os usuários especialistas são pessoas que desenvolvem e operam a infraestrutura de TI, podendo estar dedicados na operação de hardwares ou softwares [6], [56].

A caracterização dos usuários descrita compatibiliza com as abordagens decorrentes da ISO/IEC 25010:2014 [12]. A Norma aborda a dimensão da qualidade do produto a partir de um conjunto de características onde encontramos o atributo da usabilidade que, permitirá a este trabalho manter a coesão entre conceito e experiências consolidadas via a Norma ISO/IEC.

Dessa forma ao identificar nos recursos humanos os elementos que contribuem para o aprisionamento, observa-se uma clara ligação entre os tipos de usuários e os efeitos que os mesmos causam na infraestrutura de TI. No tocante a operação dos componentes da infraestrutura, pelos usuários especialistas, ou seja pela utilização dos resultados gerados externamente por esta infraestrutura para os usuários finais.

Entretanto, como a Norma [12] trata de software, o presente trabalho mesclou elementos identificados nos recursos de hardware e software para contemplar os usuários especialistas

dedicados ao hardware, especialistas dedicados ao software e usuários finais. Assim, será apresentado na Tabela 3.5, da subseção 3.4.3, a existência de elementos que contribuem para elevação do aprisionamento a partir dos três tipos de usuários.

### 3.4 Proposta de índice de aprisionamento na infraestrutura de TI

O presente trabalho buscou elencar elementos que pudessem orientar, verificar e medir o aprisionamento na infraestrutura de TI em diversas situações.

Dessa forma, o presente guia é composto por 13 elementos tendo a seguinte distribuição: 3 elementos no recurso de hardware; 6 elementos no recurso de software e; 4 elementos nos recursos humanos. Cada elemento abre para descrições específicas correspondente às características do próprio elemento e em cada descrição é atribuída uma pergunta para estabelecer o significado do elemento com o aprisionamento.

Os elementos, suas descrições e perguntas estão organizados em 3 tabelas correspondentes a cada grupo de recursos avaliados, sendo que as tabelas possuem a mesma estrutura. As tabelas estão divididas em 4 colunas, sendo: elemento, descrição, questões apontadas, peso e nota. Cabe destacar as seguintes explicações:

- **coluna de peso** - foram propostos pesos a partir da relevância de cada elemento no contexto do aprisionamento sendo que na somatória fosse fixado o valor entre 0 e 1. Entretanto, para algumas descrições o peso extrapola o máximo de 1 a fim de viabilizar na nota um quadro mais real de cada elemento e ou descrição.
- **coluna de nota** - foi proposto o intervalo de nota de 1 a 10 sendo 1 a pior e 10 a melhor avaliação. Para algumas descrições a nota assume valor maior que 10 a fim de possibilitar uma maior relevância para valores que distanciem do aprisionamento.

Por fim, a relação entre o peso e a nota atribuída, resultará no índice que será apresentado na seção 3.5.

#### 3.4.1 Parte 1 - Elementos que compõem os recursos de hardware

Para o recurso de hardware foram identificados 3 elementos, 6 descrições e 8 questões apontadas, conforme a Tabela abaixo:

Tabela 3.3: Elementos dos recursos de hardware considerados críticos ao aprisionamento

Fonte: o autor

Elemento	Descrição	Questões apontadas	Peso	Nota (1-10)
Armazenamento não primário	Mesma marca	Existe espaço para expansão da capacidade? Onde 1=inexiste; 10=existe totalmente.	0.2	
	Marcas diferentes	Existe uma camada de integração entre os elementos do mesmo tipo e de marcas diferentes? Onde 1=inexiste; 10=existe totalmente.	0.1	

*Continua na próxima página*

Tabela 3.3 – Continuação

Elemento	Descrição	Questões apontadas	Peso	Nota (1-10)
		Existe espaço para expansão da capacidade? Onde 1=ine-existe; 10=existe totalmente.	0.1	
Armazenamento primário (RAM)	Mesma marca	Existe espaço para expansão da capacidade? Onde 1=ine-existe; 10=existe totalmente.	0.5	
	Marcas diferentes	Existe uma camada de integração entre os elementos do mesmo tipo e de marcas diferentes? Onde 1=inexiste; 10=existe totalmente.	0.3	
		Existe espaço para expansão da capacidade? Onde 1=ine-existe; 10=existe totalmente.	0.3	
Processadores / microprocessadores	Mesma marca e geração	Existe espaço para expansão da capacidade? Onde 1=ine-existe; 10=existe totalmente.	0.5	
	Capacidade	A frequência (o Clock) com que o processador consegue executar a tarefa está compatível com os demais elementos do hardware? Onde 1=incompatível; 10=compatível totalmente.	1	

Considerou-se, para elaboração do quadro acima, que tanto para as memórias primária e não primárias é possível a convivência de marcas diferentes no mesmo conjunto de hardware, cabendo a recomendação que para o processamento considerado crítico a situação ideal é que todos os discos ou pentes de memória sejam da mesma marca.

Em relação aos demais elementos do recurso de hardware, cabe considerar a necessidade de uma observação maior a fim de detectar a existência ou não de mais elementos que possam contribuir significativamente para o aprisionamento. Nesse caso, no momento em que irá preparar a aplicação, deverão ser observadas as duas questões apresentadas anteriormente:

- 1 - qual a finalidade ou o propósito do hardware avaliado ou a ser adquirido?
- 2 - o propósito a que se destina é geral ou específico?

### 3.4.2 Parte 2 - Elementos que compõem os recursos de software

A tabela abaixo, para verificação dos recursos de software, é composta de 6 elementos, 21 descrições e 23 questões apontadas:



Tabela 3.4: Elementos dos recursos de software considerados críticos ao aprisionamento

Fonte: o autor

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
Usabilidade	Reconhecibilidade	Quanto que o usuário consegue reconhecer se o produto é adequado às suas necessidades? Onde 1=não reconhece; 10=reconhece totalmente.	0.2	
	Apreensibilidade	Qual a capacidade que o software possui para alcançar objetivos de aprendizagem, garantindo a satisfação do usuário? Onde 1=insatisfeito, 10=satisfeito totalmente.	0.2	
	Operacionalidade	Quanto que o software permite ao usuário operá-lo e controlá-lo de forma fácil? Onde 1=não permite; 10=permite totalmente.	0.2	
	Estética da Interface	Quanto da interface é satisfatória para o usuário? Onde 1=insatisfatória; 10=satisfaz totalmente.	0.2	
	Acessibilidade	Qual a capacidade de ser usado por usuários com diferentes características e habilidades para alcançar objetivos específicos? Onde 1=incapaz; 10=capaz totalmente.	0.2	
Compatibilidade	Coexistência	Quanto é compartilhado de recursos com outro software sem causar impactos sobre qualquer outro produto? Onde 1=não compartilha; 10=compartilha totalmente.	0.5	
	Interoperabilidade	Qual a capacidade de interagir com um ou mais sistemas especificados? Onde 1=não interage; 10=interage totalmente.	0.5	
Manutenibilidade	Modularidade	Qual o esforço para alterar elementos do software com impacto mínimo? Onde 1=muito esforço; 10=baixo esforço.	1	

*Continua na próxima pagina*

Tabela 3.4 – *Continuação*

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
	Reutilização	Quantos componentes do software podem ser utilizados por outros sistemas existentes ou em construção? Onde 1=nada utilizado; 10=utilizados totalmente.	10	
	Analisabilidade	Qual o esforço para avaliar o impacto de uma mudança em uma ou em mais partes do sistema? Qual esforço para diagnosticar partes do sistema em que podem haver falhas e identificar componentes a serem modificados? Onde 1=muito esforço; 10=baixo esforço.	1	
		Qual o grau de documentação do software? Onde 1=baixo grau; 10=alto grau.	1	
	Modificabilidade	Quanto é possível implementar uma modificação sem causar defeitos no produto existente? Onde 1=impossível; 10=altamente possível.	1	
Portabilidade	Adaptabilidade	Quanto pode ser adaptado para ambientes de operação especificados, sem a necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo software considerado? Onde 1=nenhuma adaptação; 10=altamente adaptável.	0.3	
	Instalabilidade	Qual o esforço para instalar e/ou desinstalar corretamente em um ambiente especificado? Onde 1=nível alto de esforço; 10=nível baixo de esforço.	0.35	
	Substituibilidade	Qual o esforço para substituir outro software no mesmo ambiente e para o mesmo fim? Onde 1=nível alto de esforço; 10=nível baixo de esforço	0.35	
Se utilizou framework / pacotes	Maturidade da comunidade	Quantos anos existe a comunidade? Onde 1=um ano; 10=5 anos ou mais.	3	

*Continua na próxima página*

Tabela 3.4 – Continuação

Elemento	Descrição	Questões apontadas	Peso	Nota (1-10)
/ scripts disponibilizados por comunidades de desenvolvedores	Estabilidade da comunidade	Qual o número de desenvolvedores atuantes? Onde 1=3 desenvolvedores; 10=30 ou mais desenvolvedores.	30	
	Suporte pago?	Quanto da solução utilizada para solucionar o problema é transferido ao longo do suporte? Onde 1=nada transferido; 10=transferido totalmente.	3	
	Suporte por fórum?	Qual o tempo de resposta do fórum à demanda levantada? Onde 1=muito tempo; 10=retorno imediato.	1	
Categoria de licença de software	Permissiva	Quantas licenças são da categoria permissiva? Onde 1=nenhuma, 10=todas.	10	
	Recíprocas totais	Quantas licenças são da categoria recíprocas totais? Onde 1=nenhuma, 10=todas.	20	
	Recíprocas parciais	Quantas licenças são da categoria recíprocas parciais? Onde 1=nenhuma, 10=todas.	30	

Como constado na tabela acima, algumas descrições possuem pesos que extrapolam o peso máximo de 1. Para essas descrições, cabem os seguintes comentários:

- reutilização - Como a pergunta indica a quantidade de componentes de software em números absolutos, é necessário extrapolar o peso a fim de liberar ao avaliador a possibilidade de indicar o real número de componentes reutilizáveis. Pois nessa descrição, quanto maior o número, mais a descrição irá contribuir para um menor aprisionamento.
- maturidade da comunidade - considerou-se que quanto maior o tempo, em anos, menor será a possibilidade de a comunidade de desenvolvedores fechar.
- estabilidade da comunidade - foi estabelecido o número de 30 desenvolvedores como item de estabilidade a partir da experiência do Centro de Computação Científica e Software Livre (C3SL) que possui intensa relação com a comunidade de desenvolvedores.
- suporte pago - esse suporte pago considera o suporte fornecido tanto por empresas desenvolvedoras de software, quanto por comunidades que tem como prática a divulgação gratuita do código fonte do software.

- categoria de licença de software - foi alterado o peso pertinente ao tipo de categoria a fim de possibilitar ao avaliador a contagem em número absoluto de licenças. Nesse caso, será considerado menos aprisionado quanto maior for o número de licenças na categoria permissiva, e para mais aprisionado quanto maior for o número para categoria recíprocas parciais.

### 3.4.3 Parte 3 - Elementos que compõem os recursos humanos

Segue abaixo a proposta de tabela para avaliação dos elementos que compõem os recursos humanos:

Tabela 3.5: Elementos dos recursos humanos considerados críticos ao aprisionamento

Fonte: o autor

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
Especialista dedicado ao hardware	marca do hardware	Quanto que o especialista domina dos detalhes críticos pertinentes a marca existente? Onde 1=não domina; 10=domina totalmente.	0.2	
	marca diferente	Quanto que o especialista domina o conhecimento básico crítico pertinente a outras marcas existentes? Onde 1=não domina; 10=domina totalmente.	0.4	
	camada de integração	Quanto que o especialista domina a camada de integração? Onde 1=não domina; 10=domina totalmente.	0.4	
Especialista dedicado ao software	Paradigma de desenvolvimento	Quanto que o especialista domina do paradigma utilizada no desenvolvimento? Onde 1=não domina; 10=domina totalmente.	1	
	Linguagem de programação	Quanto que o especialista domina da linguagem utilizado no desenvolvimento? Onde 1=não domina; 10=domina totalmente.	1	
	Modularidade	Qual experiência para alterar elementos do software com impacto mínimo? Onde 1=inexperiente; 10=experiente totalmente.	1	

*Continua na próxima página*

Tabela 3.5 – *Continuação*

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
	Reutilização	Quanto de experiência possui na reutilização de componentes do software por outros sistemas existentes ou em construção? Onde 1=nada experiente; 10=muito experiente.	1	
	Analísabilidade	Qual a experiência em análises de impacto de uma mudança em uma ou mais partes do sistema? Qual a experiência em diagnosticar partes do sistema em que podem haver falhas e identificar componentes a serem modificados? Onde 1=inexperiente; 10=experiente totalmente.	1	
	Modificabilidade	Qual a experiência em implementar uma modificação sem causar defeitos no produto existente? Onde 1=inexperiente; 10=experiente totalmente.	1	
	Testabilidade	Qual a experiência em estabelecer critérios de teste para um sistema que foi modificado e de determinar se esses critérios foram cumpridos de forma eficaz e eficiente? Onde 1=inexperiente; 10=experiente totalmente.	1	
	Usabilidade	Qual a experiência em desenvolver software com alta usabilidade? Onde 1=inexperiente; 10=experiente totalmente.	1	
	Compatibilidade	Qual a experiência em desenvolver software com alta compatibilidade? Onde 1=inexperiente; 10=experiente totalmente.	1	
	Portabilidade	Qual a experiência em desenvolver software com alta portabilidade? Onde 1=inexperiente; 10=experiente totalmente.	1	

*Continua na próxima página*

Tabela 3.5 – *Continuação*

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
	licença de software	Qual o domínio do especialista em relação as licenças utilizadas no software? Onde 1=nenhum domínio; 10=domina totalmente.	1	
Usuário típico	Reconhecibilidade	Qual a experiência em reconhecer se o produto é adequado às suas necessidades? Onde 1=inexperiente; 10=experiente totalmente. Ou seja, o usuário consegue perceber as suas necessidades que serão espelhadas no software? Onde 1=não percebe; 10=percebe totalmente.	1	
	Apreensibilidade	Qual a experiência do usuário em reconhecer que o software pode alcançar os objetivos de aprendizagem necessárias a sua tarefa? Onde 1=inexperiente; 10=experiente totalmente.	1	
	Operacionalidade	Quanto que o usuário consegue operar e controlar o sistema de forma fácil? Onde 1=não opera nem controla; 10= opera e controla totalmente.	1	
	Estética e interface	Quanto que o usuário percebe que a interface o satisfaz? Onde 1= não percebe; 10= percebe totalmente.	1	
	Acessibilidade	Quantos usuários com diferentes características e habilidades usaram o sistema? Onde 1=nenhum; 10= mais de 10 usuários.	1	
		Qual a capacidade desses usuários, com diferentes características e habilidades, em perceber o alcance dos objetivos pretendidos? Onde 1=incapaz; 10=capaz totalmente.	1	

*Continua na próxima pagina*



Tabela 3.5 – Continuação

Elemento	Descrição	Questões apontadas	Peso	Nota (1-10)
Usuário especialista	Estética da interface	Quanto o usuário consegue reconhecer que uma interface seja satisfatório ao propósito estabelecido? Onde 1=não reconhece; 10=reconhece totalmente.	0.4	
	Reconhecibilidade	Quanto o usuário consegue reconhecer se produto é adequado às suas necessidades? Onde 1=não reconhece; 10=reconhece totalmente.	0.6	

Para os recursos humanos a avaliação deve ser segmentada em recursos humanos especialistas, sendo especialistas os dedicados ao hardware e dedicados ao software, e usuários típicos. Essa divisão é necessária pois são diferentes os níveis de solução.

### 3.5 Proposta de metodologia para aplicação do índice

Como a aplicação do Índice está dividido em três tabelas, seguindo os três blocos de recursos - hardware, software e recursos humanos - da infraestrutura de TI, sugere-se a seguinte proposta de aplicação:

- Iniciar pela conferência dos elementos dos recursos a fim de verificar a necessidade de incluir ou excluir elementos e;
- Considerar as seguintes sugestões para:
  - Recurso de hardware:
    - \* **1** - qual a finalidade ou o propósito do hardware avaliado ou a ser adquirido?
    - \* **2** - o propósito a que se destina é geral ou específico?
- Recurso de software: tanto nos casos de avaliar um software específico ou realizar a consolidação do inventário dos softwares gerenciados na infraestrutura de TI, deve-se separar entre software de padrão fechado e software de padrão aberto. Essa separação irá alterar o índice otimizando o processo de avaliação.
- Recursos humanos: sempre que possível envolver de forma articulada as figuras dos usuários, sendo elas o usuário especialista de hardware, especialista de software e o usuário típico.

### 3.6 Cálculo do índice de aprisionamento na infraestrutura de TI

Para o cálculo de aprisionamento da infraestrutura ( $\Delta$ ), serão somados os valores obtidos nas tabelas 3.3, 3.4 e 3.5 e dividido por 3.

$$\Delta = \frac{\Gamma + \frac{\sum_{k=1}^n \Upsilon_k}{n} + \Psi}{3} \quad (3.1)$$

Onde  $\Gamma$  representa os valores do hardware,  $\Upsilon$  são os valores de todos os softwares e  $\Psi$  significa os recursos humanos.

Para solucionar as tabelas mencionadas, todos os pesos devem ser divididos pelas notas

$$Resultado = \frac{peso}{nota} \quad (3.2)$$

Resolvendo a tabela 3.3:

$$\Gamma = \frac{\gamma_1 + \gamma_2 + \gamma_3}{3} \quad (3.3)$$

Onde  $\gamma_1$  é Armazenamento não primário,  $\gamma_2$  Armazenamento primário(RAM) e  $\gamma_3$  Processadores/microprocessadores

Resolvendo a tabela 3.4:

$$\Upsilon = \frac{v_1 + v_2 + v_3 + v_4 + v_5 + v_6}{n} \quad (3.4)$$

Onde:

- $v_1$  é Usabilidade
- $v_2$  Compatibilidade
- $v_3$  Manutenibilidade
- $v_4$  Portabilidade
- $v_5$  Se utilizou framework / pacotes / scripts disponibilizados por comunidades de desenvolvedores
- $v_6$  Categoria de licença de software
- $n$  a quantidade de elementos considerados

Para os elementos considerados, aplicar a equação 3.5:

$$v = \frac{\sum_{k=1}^i resultado_i}{\sum_{k=1}^i peso_i} \quad (3.5)$$

Resolvendo a tabela 3.5:

$$\Psi = \frac{\psi_1 + \psi_2 + \psi_3 + \psi_4}{n} \quad (3.6)$$

Onde:

- $\psi_1$  é especialista dedicado ao hardware

- $\psi_2$  especialista dedicado ao software
- $\psi_3$  usuário típico
- $\psi_4$  usuário especialista
- $n$  a quantidade de elementos considerados

Para os elementos considerados, aplicar a equação 3.7:

$$v = \frac{\sum_{k=1}^i resultado_i}{\sum_{k=1}^i peso_i} \quad (3.7)$$

As equações 3.3, 3.4 e 3.6 estão no intervalo de 0 a 1, logo a equação 3.1 também está no intervalo 0 a 1, onde 1 é aprisionamento máximo.

## 4 Estudos de caso

Neste capítulo será aplicado e calculado o índice em dois estudos de caso escolhidos de forma aleatória, porém observando os critérios de existência real para efeito de conferência, disponibilização pública das documentações utilizadas para fundamentar os casos e abrangência. Buscou-se utilizar casos que demonstrassem exemplos próximos do menor e do maior nível de aprisionamento.

### 4.1 1ª Aplicação do índice

Um caso de aquisição de dispositivo móvel

#### 4.1.1 Objetivo

Identificar no Edital do Pregão Eletrônico nº: 63/2015 da Procuradoria Geral da República a existência de elementos dos recursos da infraestrutura de TI e o nível de aprisionamento oriundo da escolha descrita no referido Edital. Será buscada resposta para as seguintes questões:

- **Decisão** - qual foi a fundamentação para a presente escolha da tecnologia?
- **Recursos da infraestrutura de TI** - qual recuso da infraestrutura concentrou o índice de aprisionamento?
- **Índice de aprisionamento** - Qual o valor do índice de aprisionamento gerado?
- **Custo de troca** - o custo para futura troca da tecnologia será baixo ou alto?

#### 4.1.2 Descrição do caso

Trata-se da análise de aparelhos de dispositivos móveis, smartphones fornecidos via o Edital do Pregão Eletrônico nº: 63/2015 da Procuradoria Geral da República - PGR, que tem como propósito o registro de preços para contratação de empresa especializada para prestação de serviços de telecomunicações de voz e dados, onde os dispositivos móveis mencionados entram em regime de comodato. O detalhamento do smartphone está descrito no Termo de Referência (Anexo I do Edital), mais especificamente na parte V - Especificações.

Nas especificações a PGR descreveu o tipo de smartphone desejado para aquisição, conforme Figura abaixo:

**TABELA 4 – Aparelhos Categoria 1 – Smartphones tipo A**

Aparelho: APPLE – IPHONE 6 OU SUPERIOR	
Item	Especificações Técnicas – Configuração Básica
Tecnologia	<b>3G:</b> HSDPA 850 / 900 / 1900 / 2100; <b>4G:</b> LTE ou superior
Sistema Operacional	iOS 8 ou superior
Display	Tela Multi-Touch de 4,7 polegadas
Bateria	Bateria: 1.810 mAh, Autonomia de conversação de 840 minutos
Memória de Armazenamento	64 GB
Conectividade	Através de Cabo para conexão com o PC (incluso no kit), WiFi 802.11ac, Bluetooth 4.0, NFC, 4G LTE
GPS	GPS Assistido (A-GPS)
Transmissão de Dados	4G, 3G, HSDPA, EDGE
Câmera	Câmera iSight de 8 megapixels com pixels de 1,5µ
Processador	Processador/Chipset: Apple A8 dual-core de 2 GHz
Dimensões	13,75 cm x 6,7 cm
Peso	113 gramas aproximadamente
Espessura	0,7 cm

Figura 4.1: Descrição do hardware e do software segundo o edital. Fonte:[11]

#### 4.1.3 Aplicação do índice

Na tabela de recursos de hardware foram aplicadas as notas considerando que o smartphone da marca *Apple* não possibilita qualquer expansão de memória, muito menos convive com algum tipo de hardware de outra marca.

Como é de conhecimento público, o desempenho na relação processamento e velocidade é um item positivo nos equipamentos da *Apple*. O que ficou constatado na Tabela 4.1.

Tabela 4.1: 1ª Aplicação - elementos dos recursos de hardware considerados críticos ao aprisionamento

Fonte: o autor

Elemento	Descrição	Questões apontadas	Peso	Nota (1-10)
Armazenamento não primário	Mesma marca	Existe espaço para expansão da capacidade? Onde 1=ine-existe; 10=existe totalmente.	0.2	1
	Marcas diferentes	Existe uma camada de integração entre os elementos do mesmo tipo e de marcas diferentes? Onde 1=inexiste; 10=existe totalmente.	0.1	1
		Existe espaço para expansão da capacidade? Onde 1=ine-existe; 10=existe totalmente.	0.1	1

Continua na próxima página

Tabela 4.1 – *Continuação*

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
Armazenamento primário (RAM)	Mesma marca	Existe espaço para expansão da capacidade? Onde 1=ine-existe; 10=existe totalmente.	0.5	1
	Marcas diferentes	Existe uma camada de integração entre os elementos do mesmo tipo e de marcas diferentes? Onde 1=ine-existe; 10=existe totalmente.	0.3	1
		Existe espaço para expansão da capacidade? Onde 1=ine-existe; 10=existe totalmente.	0.3	1
Processadores / microprocessadores	Mesma marca e geração	Existe espaço para expansão da capacidade? Onde 1=ine-existe; 10=existe totalmente.	0.3	1
	Capacidade	A frequência (o Clock) com que o processador consegue executar a tarefa está compatível com os demais elementos do hardware? Onde 1=incompatível; 10=compatível totalmente.	1	7,0

Em se tratando do software, na aplicação da Tabela 4.2 foi considerado sempre o pior cenário para as características internas, pois o software é proprietário, não permitindo qualquer alteração na parte estrutural do sistema operacional ou até de outros aplicativos.

Ainda em relação ao software no elemento de usabilidade, as notas foram consideravelmente elevadas tendo em vista os investimentos que a marca realiza na característica da usabilidade, conforme Tabela 4.2, abaixo.

Tabela 4.2: 1ª Aplicação - elementos dos recursos de software considerados críticos ao aprisionamento

Fonte: o autor

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
Usabilidade	Reconhecibilidade	Quanto o usuário consegue reconhecer se o produto é adequado às suas necessidades? Onde 1=não reconhece; 10=reconhece totalmente.	0.2	9,5

*Continua na próxima página*



Tabela 4.2 – *Continuação*

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
	Apreensibilidade	Qual a capacidade que o software possui para alcançar objetivos de aprendizagem, garantindo a satisfação do usuário? Onde 1=insatisfeito, 10=satisfeito totalmente.	0.2	8,7
	Operacionalidade	Quanto o software permite ao usuário operá-lo e controlá-lo de forma fácil? Onde 1=não permite; 10=permite totalmente.	0.2	9,0
	Estética da Interface	Quanto da interface é satisfatória para o usuário? Onde 1=insatisfatória; 10=satisfaz totalmente.	0.2	9,5
	Acessibilidade	Qual a capacidade de ser usado por usuários com diferentes características e habilidades para alcançar objetivos específicos? Onde 1=incapaz; 10=capaz totalmente.	0.2	9,5
Compatibilidade	Coexistência	Quanto é compartilhado de recursos com outro software sem causar impactos sobre qualquer outro produto? Onde 1=não compartilha; 10=compartilha totalmente.	0.5	2,0
	Interoperabilidade	Qual a capacidade de interagir com um ou mais sistemas especificados? Onde 1=não interage; 10=interage totalmente.	0.5	1
Manutenibilidade	Modularidade	Qual o esforço para alterar elementos do software com impacto mínimo? Onde 1=muito esforço; 10=baixo esforço.	1	1
	Reutilização	Quantos componentes do software podem ser utilizados por outros sistemas existentes ou em construção? Onde 1=nada utilizado; 10=utilizados totalmente.	10	1,0

*Continua na próxima pagina*

Tabela 4.2 – *Continuação*

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
	Analisabilidade	Qual o esforço para avaliar o impacto de uma mudança em uma ou em mais partes do sistema? Qual esforço para diagnosticar partes do sistema em que podem haver falhas e identificar componentes a serem modificados? Onde 1=muito esforço; 10=baixo esforço.	1	1
		Qual o grau de documentação do software? Onde 1=baixo grau; 10=alto grau.	1	1
	Modificabilidade	Quanto é possível implementar uma modificação sem causar defeitos no produto existentes? Onde 1=impossível; 10=altamente possível.	1	1
Portabilidade	Adaptabilidade	Quanto pode ser adaptado para ambientes de operação especificados, sem a necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo software considerado? Onde 1=nenhuma adaptação; 10=altamente adaptável.	0.3	1
	Instalabilidade	Qual o esforço para instalar e/ou desinstalar corretamente em um ambiente especificado? Onde 1=nível alto de esforço; 10=nível baixo de esforço.	0.35	1
	Substituibilidade	Qual o esforço para substituir outro software no mesmo ambiente e para o mesmo fim? Onde 1=nível alto de esforço; 10=nível baixo de esforço.	0.35	5
Se utilizou framework / pacotes / scripts disponibilizados por comunidades	Maturidade da comunidade	Há quantos anos existe a comunidade? Onde 1=um ano; 10=5 anos ou mais.	3	1
	Estabilidade da comunidade	Qual o número de desenvolvedores atuantes? Onde 1=3 desenvolvedores; 10=30 ou mais desenvolvedores.	30	1

*Continua na próxima página*

Tabela 4.2 – Continuação

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
de desenvolvedores	Suporte pago?	Quanto da solução utilizada para solucionar o problema é transferido ao longo do suporte? Onde 1=nada transferido; 10=transferido totalmente.	3	5
	Suporte por fórum?	Qual o tempo de resposta do fórum à demanda levantada? Onde 1=muito tempo; 10=retorno imediato.	1	1
Categoria de licença de software	Permissiva	Quantas licenças são da categoria permissiva? Onde 1=nenhuma, 10=todas.	10	1
	Recíprocas totais	Quantas licenças são da categoria recíprocas totais? Onde 1=nenhuma, 10=todas.	20	1
	Recíprocas parciais	Quantas licenças são da categoria recíprocas parciais? Onde 1=nenhuma, 10=todas.	30	1

Ao aplicar a Tabela 4.3 referente aos recursos humanos, foi considerado para o usuário especialista, seja em hardware ou em software, a nota 5 quando a pergunta se referir ao conhecimento. Essa decisão foi baseada no entendimento de que mesmo conhecendo-os, o usuário especialista não estaria autorizado a modificar os recursos de infraestrutura.

Tabela 4.3: 1ª Aplicação - elementos dos recursos humanos considerados críticos ao aprisionamento

Fonte: o autor

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
Especialista dedicado ao hardware	marca do hardware	Quanto o especialista domina dos detalhes críticos pertinentes à marca existente? Onde 1=não domina; 10=domina totalmente.	0.2	3
	marca diferente	Quanto o especialista domina o conhecimento básico crítico pertinente a outras marcas existentes? Onde 1=não domina; 10=domina totalmente.	0.4	3

*Continua na próxima página*

Tabela 4.3 – *Continuação*

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
	camada de integração	Quanto o especialista domina a camada de integração? Onde 1=não domina; 10=domina totalmente.	0.4	1
Especialista dedicado ao software	Paradigma de desenvolvimento	Quanto o especialista domina do paradigma utilizada no desenvolvimento? Onde 1=não domina; 10=domina totalmente.	1	5
	Linguagem de programação	Quanto o especialista domina da linguagem utilizada no desenvolvimento? Onde 1=não domina; 10=domina totalmente.	1	5
	Modularidade	Qual experiência para alterar elementos do software com impacto mínimo? Onde 1=inexperiente; 10=experiente totalmente.	1	1
	Reutilização	Quanto de experiência possui na reutilização de componentes do software por outros sistemas existentes ou em construção? Onde 1=nada experiente; 10=muito experiente.	1	1
	Analisabilidade	Qual a experiência em análises de impacto de uma mudança em uma ou mais partes do sistema? Qual a experiência em diagnosticar partes do sistema em que podem haver falhas e identificar componentes a serem modificados? Onde 1=inexperiente; 10=experiente totalmente.	1	5
	Modificabilidade	Qual a experiência em implementar uma modificação sem causar defeitos no produto existente? Onde 1=inexperiente; 10=experiente totalmente.	1	1

*Continua na próxima pagina*

Tabela 4.3 – *Continuação*

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
	Testabilidade	Qual a experiência em estabelecer critérios de teste para um sistema que foi modificado e de determinar se esses critérios foram cumpridos de forma eficaz e eficiente? Onde 1=inexperiente; 10=experiente totalmente.	1	5
	Usabilidade	Qual a experiência em desenvolver software com alta usabilidade? Onde 1=inexperiente; 10=experiente totalmente.	1	5
	Compatibilidade	Qual a experiência em desenvolver software com alta compatibilidade? Onde 1=inexperiente; 10=experiente totalmente.	1	5
	Portabilidade	Qual a experiência em desenvolver software com alta portabilidade? Onde 1=inexperiente; 10=experiente totalmente.	1	4
	licença de software	Qual o domínio do especialista em relação as licenças utilizadas no software? Onde 1=nenhum domínio; 10=domina totalmente.	1	3
Usuário típico	Reconhecibilidade	Qual a experiência em reconhecer se o produto é adequado às suas necessidades ? Ou seja, o usuário consegue perceber as suas necessidades que serão espelhadas no software? Onde 1=não percebe; 10=percebe totalmente.	1	7
	Apreensibilidade	Qual a experiência do usuário em reconhecer que o software pode alcançar os objetivos de aprendizagem necessárias à tarefa? Onde 1=inexperiente; 10=experiente totalmente.	1	8

*Continua na próxima pagina*

Tabela 4.3 – *Continuação*

Elemento	Descrição	Questões apontadas	Peso	Nota (1-10)
	Operacionalidade	Quanto o usuário consegue operar e controlar o sistema de forma fácil? Onde 1=não opera nem controla; 10= opera e controla totalmente.	1	8
	Estética e interface	Quanto o usuário percebe que a interface o satisfaz? Onde 1= não percebe; 10= percebe totalmente.	1	9
	Acessibilidade	Quantos usuários com diferentes características e habilidades usaram o sistema? Onde 1=nenhum; 10= mais de 10 usuários.	1	8
		Qual a capacidade desses usuários, com diferentes características e habilidades, em perceber o alcance dos objetivos pretendidos? Onde 1=incapaz; 10=capaz totalmente.	1	5
Usuário especialista	Estética da interface	Quanto o usuário consegue reconhecer que uma interface seja satisfatório ao propósito estabelecido? Onde 1=não reconhece; 10=reconhece totalmente.	0.4	7
	Reconhecibilidade	Quanto o usuário consegue reconhecer se o produto é adequado às suas necessidades? Onde 1=não reconhece; 10=reconhece totalmente.	0.6	6

#### 4.1.4 Cálculo do índice de aprisionamento na aquisição

Conforme demonstrado na tabela 4.1, os valores são:

$\gamma_1 = 0.4$ ,  $\gamma_2 = 1.1$  e  $\gamma_3 = 0.44$ , aplicando a equação 3.3 temos  $\Gamma = 0.64$

Conforme demonstrado na tabela 4.2, os valores são:

$\nu_1 = 1684.04$ ,  $\nu_2 = 11655$ ,  $\nu_3 = 15540$ ,  $\nu_4 = 11188,80$ ,  $\nu_5 = 15540$  e  $\nu_6 = 15540$ , aplicando a equação 3.4 temos  $\Upsilon = 0.763$

Conforme demonstrado na tabela 4.3, os valores são:

$\psi_1 = 39.6$ ,  $\psi_2 = 28.7$ ,  $\psi_3 = 9.11$  e  $\psi_4 = 10.37$ , aplicando a equação 3.6 temos  $\Psi = 0.33$

Jogando os valores obtidos na equação 3.1, temos  $\Delta = 0.58$  sendo o índice de aprisionamento.



### 4.1.5 Considerações

Como resultado da aplicação do índice para o caso de aquisição de hardware do tipo smartphone podemos considerar:

- **Decisão** - qual foi a fundamentação para a presente escolha da tecnologia?

Nota-se no Edital do Pregão Eletrônico da PGR que uma das justificativas foi a aquisição anterior de iPads, que motivou novamente a aquisição de outro hardware da mesma fabricante. Essa argumentação explícita no documento oficial corrobora a literatura de que após a primeira decisão de compra o ciclo do aprisionamento irá diminuir as possibilidades de escolha em uma futura aquisição.

- **Recursos da infraestrutura de TI** - qual recurso da infraestrutura concentrou o índice de aprisionamento?

O recurso que mais contribuiu para o aprisionamento foi o software com 0.76 seguido do hardware com 0.64. Esses valores demonstram a existência de um hardware fechado, porém com um desempenho significativo e um software também fechado - mas com um alto grau de usabilidade.

- **Custo de troca** - o custo para futura troca da tecnologia será baixo ou alto?

O guia não analisou os custos especificamente, mas podemos considerar que conforme o órgão for adquirindo novos equipamentos em algum grau ele terá que verificar a integração com os equipamentos já existentes.

- **Índice de aprisionamento** - Qual o valor do índice de aprisionamento gerado?

O índice de aprisionamento constatado foi de 0.58 demonstrando um forte equilíbrio na relação hardware e software fechados versus o desempenho e alta usabilidade. Podemos afirmar que para hardwares como smartphones e softwares dedicados a esses hardwares a relação custo benefício fica equilibrada.

## 4.2 2a Aplicação do índice

Um caso de desenvolvimento de software

### 4.2.1 Objetivo

Identificar, no Sistema Integrado de Monitoramento do Ministério da Ciência, Tecnologia, Inovação e Comunicação (SIMMCTIC) <sup>1</sup>, e na existência de elementos dos recursos da infraestrutura de TI, o nível de aprisionamento oriundo das escolhas ocorridas ao longo do desenvolvimento do Sistema. Será buscada resposta para as seguintes questões:

- **Decisão** - qual foi a fundamentação para a presente escolha da tecnologia?
- **Recursos da infraestrutura de TI** - qual recurso da infraestrutura concentrou o índice de aprisionamento?
- **Custo de troca** - o custo para futura troca da tecnologia será baixo ou alto?
- **Índice de aprisionamento** - Qual o valor do índice de aprisionamento gerado?

---

<sup>1</sup><https://simmc.c3sl.ufpr.br/>. Acesado em 20/04/2017.

#### 4.2.2 Descrição do caso

A análise ocorrerá no SIMMCTIC desenvolvido pelo grupo de pesquisa C3SL, sob demanda do Ministério da Ciência, Tecnologia, Inovação e Comunicação (MCTIC). O SIMMCTIC é um sistema de monitoramento e transparência desenvolvido para acompanhar a implementação e uso de políticas públicas de inclusão digital, presentes em cidades e regiões remotas do país, onde o investimento privado é baixo ou inexistente.

Os projetos monitorados são: a) Gesac<sup>2</sup>, que oferece conexão via satélite às escolas, unidades de saúde, vilas indígenas, quilombos, entre outros. b) Cidades Digitais<sup>3</sup>, que implementa infraestrutura de rede de alta velocidade em pequenas cidades, conectando órgãos governamentais e disponibilizando acesso gratuito em praças públicas, assim como aplicativos de e-government buscando melhorar a eficiência da administração pública. c) Telecentros<sup>4</sup>, projeto que leva para comunidades carentes um laboratório de informática com acesso a internet, equipado para realização de cursos e treinamentos, agindo como um espaço para integração, cultura e lazer [83].

O Sistema foi desenvolvido em código aberto, estando toda a documentação e código no *GitLab*, disponibilizado sob licença *Gnu General Public License* (GPL) [84] e implementado como três módulos principais: coleta, armazenamento e visualização [83].

O módulo de coleta é responsável por coletar as métricas dos dispositivos, considerando ambiente altamente heterogêneo dos projetos monitorados. O módulo de armazenamento foi implementado no modelo de *Data Warehouse* (DW) com o *PostgreSQL*<sup>5</sup> servindo como banco de dados [85]. O módulo de visualização consiste em uma interface web acessível publicamente, que oferece gráficos, relatórios e mapas, três formas distintas e complementares para visualizar e analisar os dados armazenados [83].

#### 4.2.3 Aplicação do índice

Considerando as características do sistema analisado, é preciso observar a origem do seu desenvolvimento. Essa origem está fundamentada em um grupo de pesquisa, o Centro de Computação Científica e Software Livre (C3SL), do Departamento de Informática da Universidade Federal do Paraná. O grupo existe há 16 anos e, por definição, pesquisa e desenvolve softwares ancorados em padrões de software aberto.

O centro de pesquisa possui uma infraestrutura de Datacenter que não é típica, pois se trata de uma infraestrutura dedicada à pesquisa científica. Porém, os padrões adotados podem, com ajuste de escala, ser adequados para um propósito geral e ser utilizado em organizações diversas.

Dessa forma, a análise do SIMMCTIC não destoa da realidade externa ao ambiente acadêmico, exceto pelo alto nível de especialização oriunda dos pesquisadores, o que é equilibrado pela orientação de desenvolver pesquisa de alta complexidade com resultados práticos aplicados em diferentes ambientes.

Aplicando a Tabela 4.4 referente ao recurso de hardware, identificamos uma estrutura computacional expansiva quanto à capacidade e à escalabilidade. Pode ser dado um exemplo pela configuração do hardware dedicado ao Banco de Dados que utiliza processadores Intel Xeon 3.00 GHZ, com 256GB de RAM, 6 disco em RAID 10 com 500GB utilizado.

<sup>2</sup><https://www.mctic.gov.br/mctic/opencms/comunicacao/SETEL/gesac/gesac.html> acessado em 20/04/2017.

<sup>3</sup><http://www.mctic.gov.br/mctic/opencms/indicadores/detalhe> acessado em 20/04/2017.

<sup>4</sup>[https://www.mctic.gov.br/mctic/opencms/comunicacao/SETEL/inclusao\\_digital/telecentros/TELECENROS.html](https://www.mctic.gov.br/mctic/opencms/comunicacao/SETEL/inclusao_digital/telecentros/TELECENROS.html) acessado em 20/04/2017.

<sup>5</sup><https://www.postgresql.org> acessado em 21/04/2017.

Tabela 4.4: 2ª Aplicação - elementos dos recursos de hardware considerados críticos ao aprisionamento

Fonte: o autor

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
Armazenamento não primário	Mesma marca	Existe espaço para expansão da capacidade? Onde 1=ine-existe; 10=existe totalmente.	0.2	9
	Marcas diferentes	Existe uma camada de integração entre os elementos do mesmo tipo e de marcas diferentes? Onde 1=inexiste; 10=existe totalmente.	0.1	8
		Existe espaço para expansão da capacidade? Onde 1=ine-existe; 10=existe totalmente.	0.1	8
Armazenamento primário (RAM)	Mesma marca	Existe espaço para expansão da capacidade? Onde 1=ine-existe; 10=existe totalmente.	0.5	9
	Marcas diferentes	Existe uma camada de integração entre os elementos do mesmo tipo e de marcas diferentes? Onde 1=inexiste; 10=existe totalmente.	0.3	8
		Existe espaço para expansão da capacidade? Onde 1=ine-existe; 10=existe totalmente.	0.3	8
Processadores / microprocessadores	Mesma marca e geração	Existe espaço para expansão da capacidade? Onde 1=ine-existe; 10=existe totalmente.	0.3	8
	Capacidade	A frequência (o Clock) com que o processador consegue executar a tarefa está compatível com os demais elementos do hardware? Onde 1=incompatível; 10=compatível totalmente.	1	9

Na aplicação da Tabela 4.5, nota-se como todo o desenvolvimento do sistema foi orientado ao padrão de software aberto. Fica evidente o destaque aos elementos de manutenibilidade e portabilidade mantendo o nível elevado na usabilidade.

Tabela 4.5: 2ª Aplicação - elementos dos recursos de software considerados críticos ao aprisionamento

Fonte: o autor

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
Usabilidade	Reconhecibilidade	Quanto o usuário consegue reconhecer se o produto é adequado às suas necessidades? Onde 1=não reconhece; 10=reconhece totalmente.	0.2	8
	Apreensibilidade	Qual a capacidade que o software possui para alcançar objetivos de aprendizagem, garantindo a satisfação do usuário? Onde 1=insatisfeito, 10=satisfeito totalmente.	0.2	8
	Operacionalidade	Quanto o software permite ao usuário operá-lo e controlá-lo de forma fácil? Onde 1=não permite; 10=permite totalmente.	0.2	8
	Estética da Interface	Quanto da interface é satisfatória para o usuário? Onde 1=insatisfatória; 10=satisfaz totalmente.	0.2	8
	Acessibilidade	Qual a capacidade de ser usado por usuários com diferentes características e habilidades para alcançar objetivos específicos? Onde 1=incapaz; 10=capaz totalmente.	0.2	5
Compatibilidade	Coexistência	Quanto é compartilhado de recursos com outro software sem causar impactos sobre qualquer outro produto? Onde 1=não compartilha; 10=compartilha totalmente.	0.5	9
	Interoperabilidade	Qual a capacidade de interagir com um ou mais sistemas especificados? Onde 1=incapaz; 10=capaz totalmente.	0.5	9
Manutenibilidade	Modularidade	Qual o esforço para alterar elementos do software com impacto mínimo? Onde 1=muito esforço; 10=baixo esforço.	1	9

*Continua na próxima página*

Tabela 4.5 – *Continuação*

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
	Reutilização	Quantos componentes do software podem ser utilizados por outros sistemas existentes ou em construção? Onde 1=nada utilizado; 10=utilizados totalmente.	10	9
	Analisabilidade	Qual o esforço para avaliar o impacto de uma mudança em uma ou em mais partes do sistema? Qual o esforço para diagnosticar partes do sistema em que podem haver falhas e identificar componentes a serem modificados? Onde 1=muito esforço; 10=baixo esforço.	1	8.5
		Qual o grau de documentação do software? Onde 1=baixo grau; 10=alto grau.	1	9
	Modificabilidade	Quanto é possível implementar uma modificação sem causar defeitos no produto existentes? Onde 1=impossível; 10=altamente possível.	1	7.5
Portabilidade	Adaptabilidade	Quanto pode ser adaptado para ambientes de operação especificados, sem a necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo software considerado? Onde 1=nenhuma adaptação; 10=altamente adaptável.	0.3	8.5
	Instalabilidade	Qual o esforço para instalar e/ou desinstalar corretamente em um ambiente especificado? Onde 1=nível alto de esforço; 10=nível baixo de esforço.	0.35	9
	Substituibilidade	Qual o esforço para substituir outro software no mesmo ambiente e para o mesmo fim? Onde 1=nível alto de esforço; 10=nível baixo de esforço.	0.35	7.5

*Continua na próxima pagina*

Tabela 4.5 – Continuação

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
Se utilizou framework / pacotes / scripts disponibilizados por comunidades de desenvolvedores	Maturidade da comunidade	Há quantos anos existe a comunidade? Onde 1=um ano; 10=5 anos ou mais.	3	9
	Estabilidade da comunidade	Qual o número de desenvolvedores atuantes? Onde 1=3 desenvolvedores; 10=30 ou mais desenvolvedores.	30	50
	Suporte pago?	Quanto da solução utilizada para solucionar o problema é transferido ao longo do suporte? Onde 1=nada transferido; 10=transferido totalmente.	3	10
	Suporte por fórum?	Qual o tempo de resposta do fórum à demanda levantada? Onde 1=muito tempo; 10=retorno imediato.	1	9
Categoria de licença de software	Permissiva	Quantas licenças são da categoria permissiva? Onde 1=nenhuma, 10=todas.	10	50
	Recíprocas totais	Quantas licenças são da categoria recíprocas totais? Onde 1=nenhuma, 10=todas.	20	30
	Recíprocas parciais	Quantas licenças são da categoria recíprocas parciais? Onde 1=nenhuma, 10=todas.	30	20

Na Tabela 4.6, cabe observar que a avaliação foi realizada considerando a equipe de pesquisa e desenvolvimento composta por estudantes de graduação e pós graduação, ambos orientados pelos docentes pesquisadores da universidade.

Tabela 4.6: 2ª Aplicação - elementos dos recursos humanos considerados críticos ao aprisionamento

Fonte: o autor

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
Especialista dedicado ao hardware	marca do hardware	Quanto o especialista domina dos detalhes críticos pertinentes a marca existente? Onde 1=não domina; 10=domina totalmente.	0.2	8.5

Continua na próxima página



Tabela 4.6 – *Continuação*

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
	marca diferente	Quanto o especialista domina o conhecimento básico crítico pertinente a outras marcas existentes? Onde 1=não domina; 10=domina totalmente.	0.4	8.5
	camada de integração	Quanto o especialista domina a camada de integração? Onde 1=não domina; 10=domina totalmente.	0.4	9
Especialista dedicado ao software	Paradigma de desenvolvimento	Quanto o especialista domina do paradigma utilizado no desenvolvimento? Onde 1=não domina; 10=domina totalmente.	1	9.5
	Linguagem de programação	Quanto o especialista domina da linguagem utilizada no desenvolvimento? Onde 1=não domina; 10=domina totalmente.	1	9.5
	Modularidade	Qual a experiência para alterar elementos do software com impacto mínimo? Onde 1=inexperiente; 10=experiente totalmente.	1	9
	Reutilização	Quanto de experiência possui na reutilização de componentes do software por outros sistemas existentes ou em construção? Onde 1=nada experiente; 10=muito experiente.	1	9
	Analisabilidade	Qual a experiência em análises de impacto de uma mudança em uma ou mais partes do sistema? Qual a experiência em diagnosticar partes do sistema em que podem haver falhas e identificar componentes a serem modificados? Onde 1=inexperiente; 10=experiente totalmente.	1	7

*Continua na próxima página*

Tabela 4.6 – *Continuação*

<b>Elemento</b>	<b>Descrição</b>	<b>Questões apontadas</b>	<b>Peso</b>	<b>Nota (1-10)</b>
	Modificabilidade	Qual a experiência em implementar uma modificação sem causar defeitos no produto existente? Onde 1=inexperiente; 10=experiente totalmente.	1	7.5
	Testabilidade	Qual a experiência em estabelecer critérios de teste para um sistema que foi modificado e de determinar se esses critérios foram cumpridos de forma eficaz e eficiente? Onde 1=inexperiente; 10=experiente totalmente.	1	8.5
	Usabilidade	Qual a experiência em desenvolver software com alta usabilidade? Onde 1=inexperiente; 10=experiente totalmente.	1	9
	Compatibilidade	Qual a experiência em desenvolver software com alta compatibilidade? Onde 1=inexperiente; 10=experiente totalmente.	1	8.5
	Portabilidade	Qual a experiência em desenvolver software com alta portabilidade? Onde 1=inexperiente; 10=experiente totalmente.	1	9
	licença de software	Qual o domínio do especialista em relação as licenças utilizadas no software? Onde 1=nenhum domínio; 10=domina totalmente.	1	9
Usuário típico	Reconhecibilidade	Qual a experiência em reconhecer se o produto é adequado às suas necessidades? O usuário consegue perceber as necessidades que serão espelhadas no software? Onde 1=inexperiente; 10=experiente totalmente.	1	8

*Continua na próxima página*

Tabela 4.6 – *Continuação*

Elemento	Descrição	Questões apontadas	Peso	Nota (1-10)
	Apreensibilidade	Qual a experiência do usuário em reconhecer que o software pode alcançar os objetivos de aprendizagem necessárias à tarefa? Onde 1=inexperiente; 10=experiente totalmente.	1	7
	Operacionalidade	Quanto o usuário consegue operar e controlar o sistema de forma fácil? Onde 1=não opera nem controla; 10= opera e controla totalmente.	1	8
	Estética e interface	Quanto o usuário percebe que a interface o satisfaz? Onde 1= não percebe; 10= percebe totalmente.	1	8.5
	Acessibilidade	Quantos usuários com diferentes características e habilidades usaram o sistema? Onde 1=nenhum; 10= mais de 10 usuários.	1	7
		Qual a capacidade desses usuários, com diferentes características e habilidades, em perceber o alcance dos objetivos pretendidos? Onde 1=incapaz; 10=capaz totalmente.	1	7
Usuário especialista	Estética da interface	Quanto o usuário consegue reconhecer que uma interface seja satisfatória ao propósito estabelecido? Onde 1=não reconhece; 10=reconhece totalmente.	0.4	8
	Reconhecibilidade	Quanto o usuário consegue reconhecer se produto é adequado às suas necessidades? Onde 1=não reconhece; 10=reconhece totalmente.	0.6	8

#### 4.2.4 Cálculo do índice de aprisionamento na aquisição

Conforme demonstrado na tabela 4.1, os valores são:

$\gamma_1 = 0.04$ ,  $\gamma_2 = 0.13$  e  $\gamma_3 = 0.14$ , aplicando a equação 3.3 temos  $\Gamma = 0.10$

Conforme demonstrado na tabela 4.2, os valores são:

$v_1 = 2175.60$ ,  $v_2 = 1726.66$ ,  $v_3 = 1758.58$ ,  $v_4 = 1878$ ,  $v_5 = 564.66$  e  $v_6 = 612.96$ , aplicando a equação 3.4 temos  $\Upsilon = 0.093$

Conforme demonstrado na tabela 4.3, os valores são:

$\psi_1 = 7.59$ ,  $\psi_2 = 7.66$ ,  $\psi_3 = 8.75$  e  $\psi_4 = 8.25$ , aplicando a equação 3.6 temos  $\Psi = 0.12$

Jogando os valores obtidos na equação 3.1, temos  $\Delta = 0.10$  sendo o índice de aprisionamento.

#### 4.2.5 Considerações

- **Decisão** - qual foi a fundamentação para a presente escolha da tecnologia?

O C3SL pesquisa e desenvolve software em padrão aberto tendo o código disponibilizado em repositórios públicos. Dessa forma, a definição é mais um elemento conceitual que orienta a estratégia adotada especificamente para a escolha da tecnologia em cada projeto a ser desenvolvido.

- **Recursos da infraestrutura de TI** - qual recurso da infraestrutura concentrou o índice de aprisionamento?

Foram obtidos nos três índices valores próximos de 0, demonstrando o baixo nível de aprisionamento.

- **Custo de troca** - o custo para futura troca da tecnologia será baixo ou alto?

O custo estará concentrado basicamente na manutenção corretiva e evolutiva do sistema, graças ao planejamento de desenvolvimento que considerou as subcaracterísticas da manutenibilidade.

- **Índice de aprisionamento** - Qual o valor do índice de aprisionamento gerado?

O índice de aprisionamento gerado foi de 0.10. Um índice extremamente baixo, possivelmente conquistado em decorrência das peculiaridades do grupo de pesquisa responsável pelo desenvolvimento.

## 5 Conclusão

O presente trabalho teve como objetivo propor subsídios para a identificação e quantificação do nível de aprisionamento na infraestrutura de Tecnologia da Informação (TI), considerando o processo de tomada de decisão na governança e gestão da TI.

Para tanto, foram trabalhados os conceitos de aprisionamento, governança e gestão de TI e os recursos que compõem a infraestrutura de TI. O conceito de aprisionamento tecnológico teve origem na área da Economia, mais especificamente no campo da economia da informação, que se refere tanto a informação quanto a tecnologia a ela associada [9]. No âmbito da governança corporativa das organizações, a governança de TI está contextualizada como um dos seis ativos principais que deve ser governado para que se crie valor [3].

Ao relacionar o conceito de aprisionamento com a governança corporativa e de TI, observa-se uma intrínseca relação a ponto de concluir que é no momento que as organizações tomam decisões pela mudança ou de suas estratégias, ou dos seus processos, ou das tecnologias aplicadas, que os custos de troca podem levar a uma decisão de manutenção, anulando o processo de mudança, ou seja, levando a um aprisionamento (*lock-in*) à situação atual.

Essa tomada de decisão deve estar amparada em um fluxo que considera a integração do gerenciamento da TI - que é orientado a serviços - com a governança de TI - que deve alinhar a TI com os objetivos da governança corporativa [53]. Para que essa integração ocorra de forma adequada, observou-se que os recursos da infraestrutura de TI são críticos quando relacionados ao processo de aprisionamento. Assim, conclui-se que a tomada de decisão deve considerar o estado atual dos recursos da infraestrutura para que o mesmo possa dar suporte de forma menos aprisionada aos reflexos futuros oriundos dessa decisão. Os padrões tecnológicos predominantes na infraestrutura TI indicarão o grau com que essa infraestrutura poderá ser alterada tanto em seus componentes físicos como de gestão.

Portanto, ao relacionar a tomada de decisão com a infraestrutura de TI, observa-se a ocorrência das fases do ciclo de aprisionamento. Ou seja, desde o momento da seleção da tecnologia que pode derivar para marca ou padrão tecnológico a ser executado, bem como na fase da experimentação, na fase da defesa entrincheirada pela opção tomada, culminando como o aprisionamento quando os custos de troca estão altos demais.

Esta gestão operacional da infraestrutura, envolve a escolha da tecnologia, de avaliação de riscos e incertezas, do gerenciamento de projetos e da transferência e implantação da tecnologia. Esses processos devem estar balanceados e devem contemplar estratégias que evitem o aprisionamento.

Diante do exposto, as opções de aquisição devem estar abertas durante todo o ciclo de aprisionamento. Dessa forma, identificamos autores que propõem que um número maior de opções de aquisição está associado à prospecção constante de tecnologias, à utilização de um segundo fornecedor, bem como à utilização de sistemas “abertos” e intercambiáveis. O uso de sistemas abertos significa dispor de acesso irrestrito e sem custo a códigos, documentos, especificações e tecnologias, de tal forma que seja garantida a continuidade do serviço, sem a necessidade de manter-se atrelado a um determinado fornecedor e à sua tecnologia proprietária.

Ao manter as opções abertas, a infraestrutura de TI poderá se tornar ainda mais modular e configurável, facilitando a antecipação do ciclo de aprisionamento, o que contribuirá para reduzir o nível de aprisionamento da próxima escolha.

Para ampliar o entendimento dessas características, de modular e configurável à infraestrutura de TI, o presente trabalho apresentou os recursos que compõem essa infraestrutura e optou por prosseguir a análise a partir de 3 recursos, sendo hardware, software e recursos humanos.

Para os recursos de hardware foram apresentadas algumas configurações e seus respectivos propósitos. Nessa apresentação, identificou-se a impossibilidade de demonstrar de forma exaustiva todas as possíveis configurações de hardware. Entretanto, notou-se um padrão básico que compõe uma arquitetura média dos hardwares por meio das memórias primárias (RAM) e não primárias (ROM e SSD) e os processadores, variando gerações e velocidades de processamento (*clock*). As placas de vídeos (ou aceleradores), com seus sistemas embarcados. As camadas de softwares necessárias para a comunicação com os demais softwares também apareceram com forte recorrência, porém a opção foi retirá-las da análise dos recursos de hardware devido ao seu uso extremamente dedicado.

Tendo relacionado os elementos mais críticos no recurso de hardware para identificar o aprisionamento, cabe destacar a importância do propósito ao qual o hardware será aplicado, pois esse propósito indicará uma tolerância maior ou menor em relação ao nível do aprisionamento nesse recurso.

Para a análise dos recursos de software optou-se por abstrair a localização "física" do software [59], [24], [6], tendo como o objetivo a generalização da análise, a fim de identificar os elementos comuns do recurso de software independente de sua aplicação e do hardware.

A análise buscou elementos do software que pudessem contribuir para a redução do aprisionamento. A linguagem de programação foi a primeira observada e notou-se que ao longo da sua evolução existiu uma crescente diminuição da dependência do hardware, possibilitando o uso de interface gráfica visual para programação que se aproxima mais da conversação humana. Entretanto, o domínio do paradigma da programação e da linguagem podem indicar um determinado grau de aprisionamento, focado na figura do usuário especialista - o programador.

Outro elemento analisado foi se o software possui o padrão aberto ou fechado. Este trabalho considerou o padrão fechado como a situação que mais aprisiona, tendo o seu efeito de aprisionamento reduzido nas características de usabilidade, compatibilidade e manutenibilidade. Para o padrão aberto, observou-se que mesmo os tipos de licenças com origem no movimento *Open Source* e Software Livre podem gerar um grau de aprisionamento, a exemplo dos tipos de licenças recíprocas parciais.

Ainda nos recursos de software, foram utilizadas algumas características da qualidade em uso e da qualidade do produto, extraídas da série ISO/IEC 25010:2014 [12]. A partir das características e subcaracterísticas selecionadas, pode-se concluir que a exemplo da manutenibilidade, o software é o recurso que mais contribui para aumentar ou diminuir o grau de aprisionamento na infraestrutura de TI. A reutilização, a analisabilidade e a modificabilidade só podem existir com um grau significativo da documentação do software, mas preferencialmente se combinado com o código aberto, que garantirá a transparência efetiva do processo de desenvolvimento.

Para os recursos humanos, foram utilizadas as figuras do usuário especialista dedicado ao hardware, do especialista dedicado ao software e do usuário típico. Dessa forma, pode-se concluir que a segmentação de usuários, permite o aprofundamento da avaliação conforme o nível de necessidade do elemento avaliado e do usuário disponível.

Diante da identificação dos elementos dos recursos da infraestrutura de TI que mais contribuem para a elevação do nível de aprisionamento, o presente trabalho propôs um guia composto por 13 elementos, 48 descrições e 54 perguntas, onde sugere-se uma aplicação

segmentada por recurso (hardware, software e recursos humanos), atribuindo notas para cada pergunta. A partir do guia, foi sugerido a criação de um índice de aprisionamento para a infraestrutura de TI, com o objetivo de somar individualmente cada recurso avaliado, possibilitando ainda verificar a somatória total do grau de aprisionamento, contribuindo assim com a governança e gestão de TI.

Foram selecionados dois estudos de caso para validar o guia criado. Os casos foram escolhidos de forma aleatória, porém observando os critérios de existência real para efeito de conferência, disponibilização pública das documentações utilizadas para fundamentar os casos e a abrangência através da utilização de casos que demonstraram as extremidades do intervalo 0 e 1, onde 0 é menos aprisionado e 1 significa aprisionado.

A aplicação do guia nos dois casos demonstrou a abrangência dos elementos selecionados, uma relativa versatilidade na aplicação, uma demonstração simples e objetiva dos elementos que mais contribuem para o aprisionamento e do índice geral da infraestrutura de TI.

No primeiro estudo de caso, o guia foi aplicado no edital de aquisição de um *smartphone* da marca *Apple* pela Procuradoria Geral da República (PGR). Com o índice de aprisionamento em 0,58, podemos concluir que nesse caso o aprisionamento existiu de forma moderada na soma total. Ao detalhar por recursos observou-se um grau mais elevado no recurso de hardware, com 0,64, e no recurso de software, com 0,76. Ou seja, no quesito hardware somente o elemento capacidade do processador levou uma nota próxima de 10, os demais elementos ficaram em 1, pois a marca *Apple* não possibilita expansão de memória nem a coexistência com outras marcas.

Nos elementos de software da primeira aplicação, o destaque ficou com a usabilidade, o que indicaria um custo menor para um eventual treinamento dos usuários. Só que para os demais elementos as notas atribuídas indicaram um alto grau de aprisionamento. Nos recursos humanos, excetuando os usuários especialistas, considerando o usuário típico, as notas foram elevadas, o que equilibrou o índice geral de aprisionamento.

Nessa primeira aplicação, observa-se que a opção de compra da PGR aprofundou o aprisionamento, tendo em vista que o órgão já havia adquirido hardware do tipo *ipads* e utilizou essa aquisição para justificar a nova compra.

Importante observar ainda uma relativização correspondente ao índice, na media em que o valor ficou em 0,58, para um dispositivo móvel onde a obsolescência é fator preponderante para depreciação do aparelho. A extensão do aprisionamento nessa aplicação pode ser considerada restrita, pois está sendo medida em um hardware de entrada de dados, com baixa demanda de processamento e armazenamento.

No segundo estudo de caso, o guia foi utilizado em um software de aplicação específica para o monitoramento, que utiliza um sistema de coleta distinto para cada dado coletado, com armazenamento no modelo data warehouse e um sistema de visualização.

Conforme estabelecido no presente trabalho, foram selecionados três elementos no recurso de hardware, considerados críticos para o aprisionamento, sendo eles armazenamento não primário, armazenamento primário (RAM) e processador. Dessa forma, ao analisar o recurso de hardware do sistema em questão, observou-se que a configuração necessária de hardware dos servidores para receber o sistema de monitoramento, assemelham-se aos elementos selecionados no trabalho para avaliação.

Cabe observar que no momento da análise o sistema estava hospedado no datacenter do grupo de pesquisa C3SL, o que necessita de uma avaliação específica, podendo ser desconsiderado o valor da soma de 0,10 para este recurso. Porém, a descrição do hardware apresentado em relatório de transferência do sistema ao Ministério demandante, demonstra que é possível o órgão receptor do sistema configurar um servidor que possa atender a demanda de processamento e armazenamento.



Na análise do recurso de software, a soma dos elementos conferiu 0,12, o que é considerado um sistema com baixo índice de aprisionamento. Assim, pode-se concluir pela existência de uma sincronia no processo de desenvolvimento entre os atributos internos e externos. Um forte destaque à documentação do código disponível no *GitLab* e a portabilidade que permite conviver com várias versões do *Linux* e *Windows*. Pelas características do sistema, fica constatada a limitação da nota lançada para cada pergunta, no sentido de explorar mais os fatores positivos do software, tendo em vista que o mesmo é resultado de um processo de pesquisa.

Para os elementos pertinentes aos recursos humanos, considerou-se uma nota média ao usuário especialista no sentido de aproximar o recurso humano existente em um laboratório de pesquisa e o existente no mercado. Essa consideração contribuiu para que a somatória dos elementos resultasse em 0,12, um valor que indica o baixo aprisionamento e que as tecnologias utilizadas no sistema são de domínio para além do ambiente acadêmico.

O sistema analisado demonstrou a possibilidade técnica de desenvolver um sistema inicialmente quase livre do aprisionamento, necessitando apenas da continuidade da avaliação com a finalidade de verificar esse resultado ao longo do tempo. Outro aspecto relevante refere-se ao índice geral de aprisionamento que ficou em 0,10, o menor valor apresentado para o aprisionamento na equação proposta por este trabalho.

Por fim, a partir dos referidos estudos de casos, foi possível analisar o aprisionamento de infraestrutura de tecnologia da informação como um efeito resultante dos recursos que a compõe. A consolidação do aprisionamento correspondente a somatória de todos os recursos em um índice, possibilita um apoio mais preciso ao processo de tomada de decisão da governança de TI. Com isso, espera-se contribuir para a melhoria da Governança amparada em uma infraestrutura de TI capaz de garantir a mudança de um estado atual para um estado futuro.

## 5.1 Trabalhos futuros

Como trabalhos futuros identificamos:

- Estender o campo das análises realizadas para calcular o índice do aprisionamento de serviços tecnológicos, como *Clusters* e Computação em Nuvem;
- Ampliar o índice para apoiar a aquisição de infraestrutura, deixando-o mais objetivo e com possibilidade de pontuação por parte do adquirente;
- Expandir a identificação de mais ativos da infraestrutura que podem contribuir para o aumento ou diminuição do aprisionamento.

## Referências

- [1] PAiBCP IFAC. Enterprise governance: Getting the balance right. *IFAC, New York*, 2004.
- [2] John C Henderson and Harihara Venkatraman. Strategic alignment: Leveraging information technology for transforming organizations. *IBM systems journal*, 32(1):472–484, 1993.
- [3] Peter Weill and Jeanne W Ross. *IT governance: How top performers manage IT decision rights for superior results*. Harvard Business Press, 2004.
- [4] Lowell W Steele. *Managing technology: the strategic view*. McGraw-Hill Companies, 1989.
- [5] Peter Weill and Tereza Cristina Melo de Brito Carvalho. *Governança de TI, tecnologia da informação*. M. Books, 2006.
- [6] JAMES A O'BRIEN. Sistemas de informação e as decisões na era da internet, 2ª edição. Editora Saraiva, 2004.
- [7] J.G. Brookshear. *Ciência da Computação - 11ed: Uma Visão Abrangente*. Bookman Editora, 2013.
- [8] David Beserra, Micael França, Carlos Melo, Ygor Sousa, Samuel Romeiro, Mariel Andrade, Alberto Araújo, and Erica Sousa. Ambiente virtualizado para ensino de programação paralela e computação em cluster. In *XXXIII Congresso da Sociedade Brasileira de Computação/XXI Workshop sobre Educação em Computação, Maceió*, 2013.
- [9] Carl Shapiro and Hal R Varian. *A economia da informação: como os princípios econômicos se aplicam à era da Internet*. Elsevier Brasil, 2003.
- [10] Mike Hess and Joan Enric Ricart. Managing customer switching costs: a framework for competing in the networked environment. *Management Research: Journal of the Iberoamerican Academy of Management*, 1(1):93–110, 2003.
- [11] Procuradoria Geral da República PGR. Pregão eletrônico nº: 63/2015. registro de preços. <http://www.mpf.mp.br/df/transparencia/licitacoes-contratos-e-convenios/pregoes/Editaipregao63.2015CLAROS.A.pdf>, 2016. Acessado em 30/12/2016.
- [12] Organización Internacional de Normalización. *ISO-IEC 25010: 2011 Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)-System and Software Quality Models*. ISO, 2011.
- [13] Vladimir Ferraz de ABREU and Aguinaldo Aragon FERNANDES. Implantando a governança de ti: da estratégia à gestão dos processos e serviços. *Rio de Janeiro: Brasport*, 4:656, 2014.
- [14] IBGC-INSTITUTO BRASILEIRO DE GOVERNANÇA CORPORATIVA. Código das melhores práticas de governança corporativa, 2015.

- [15] Ryan R Peterson. Integration strategies and tactics for information technology governance. *Strategies for information technology governance*, 2:37–80, 2004.
- [16] Joseph J Riggio. Understanding the sarbanes-oxley act—a valued added approach for public interest. *Critical Perspectives on Accounting*, 19(7):952–962, 2008.
- [17] Magdi R Iskander and Nadereh Chamlou. *Corporate governance: A framework for implementation*. World Bank Publications, 2000.
- [18] Peter Weill and Jeanne Ross. A matrixed approach to designing it governance. *MIT Sloan Management Review*, 46(2):26, 2005.
- [19] Didem Demirhan, Varghese S Jacob, and Srinivasan Raghunathan. Strategic it investments: the impact of switching cost and declining it cost. *Management Science*, 53(2):208–226, 2007.
- [20] Pei-yu Chen and Chris Forman. Can vendors influence switching costs and compatibility in an environment with open standards? *MIS quarterly*, pages 541–562, 2006.
- [21] Chris Forman and Pei-Yu Chen. Network effects and switching costs in the market for routers and switches. 2003.
- [22] Denis Alcides Rezende and Aline França de Abreu. Tecnologia da informação aplicada a sistemas de informação empresariais. *São Paulo: Atlas*, 9, 2013.
- [23] Herman F Hehn. Peopleware: como trabalhar o fator humano nas implementações de sistemas integrados de informação (erp). *São Paulo: Editora Gente*, 1999.
- [24] R. Pressman and B. Maxim. *Engenharia de Software - 8ª Edição*:. 2016.
- [25] Paul Coombes and Mark Watson. Three surveys on corporate governance. *McKinsey Quarterly*, (4):74–74, 2000.
- [26] Herbert Steinberg. *A dimensão humana da governança corporativa: pessoas criam as melhores e as piores práticas*. Editora Gente, 2003.
- [27] Duane Windsor. Tightening corporate governance. *Journal of International Management*, 15(3):306–316, 2009.
- [28] OECD Oecd. The oecd principles of corporate governance. *Contaduría y Administración*, (216), 2004.
- [29] Allen E Brown and Gerald G Grant. Framing the frameworks: A review of it governance research. *Communications of the Association for Information Systems*, 15(1):38, 2005.
- [30] Erik Brynjolfsson. The productivity paradox of information technology. *Communications of the ACM*, 36(12):66–77, 1993.
- [31] Erik Brynjolfsson and Lorin Hitt. Paradox lost? firm-level evidence on the returns to information systems spending. *Management science*, 42(4):541–558, 1996.
- [32] Nada Korac-Kakabadse and Andrew Kakabadse. Is/it governance: Need for an integrated model. *Corporate Governance: The international journal of business in society*, 1(4):9–11, 2001.

- [33] Kevin Amess, Sarah Brown, and Steve Thompson. Management buyouts, supervision and employee discretion. *Scottish Journal of Political Economy*, 54(4):447–474, 2007.
- [34] Sherrena Buckby, Peter Best, and Jenny Stewart. The current state of information technology governance literature. *Information Science Reference (IGI Global)*, 2008.
- [35] Ernest Jordan and David Musson. Corporate governance and it governance: Exploring the board’s perspective. 2004.
- [36] Peter Weill and Jeanne W Ross. It governance on one page. 2004.
- [37] Kenneth G Rau. Effective governance of it: design objectives, roles, and relationships. *Information Systems Management*, 21(4):35–42, 2004.
- [38] David Musson. It governance: a critical review of the literature. *Information Technology Governance and Service Management: Frameworks and Adaptations*, pages 63–80, 2008.
- [39] Peter Weill and Richard Woodham. Don’t just lead, govern: Implementing effective it governance. 2002.
- [40] Vallabh Sambamurthy and Robert W Zmud. Arrangements for information technology governance: A theory of multiple contingencies. *MIS quarterly*, pages 261–290, 1999.
- [41] Phyl Webb, Carol Pollard, and Gail Ridley. Attempting to define it governance: Wisdom or folly? In *System Sciences, 2006. HICSS’06. Proceedings of the 39th Annual Hawaii International Conference on*, volume 8, pages 194a–194a. IEEE, 2006.
- [42] Mårten Simonsson and Pontus Johnson. Defining it governance-a consolidation of literature. In *The 18th conference on advanced information systems engineering*, volume 6, 2006.
- [43] INFORMATION TECHNOLOGY GOVERNANCE INSTITUTE. Board briefing on it governance. <http://www.isaca.org>, 2003, acessado em Novembro de 2016.
- [44] Wim Van Grembergen. *Strategies for information technology governance*. Igi Global, 2004.
- [45] Fernando José Barbin Laurindo. *Tecnologia da informação: planejamento e gestão de estratégias*. Atlas, 2008.
- [46] Fernando José Barbin Laurindo, Tamio Shimizu, MM de Carvalho, and Roque Rabechini Jr. O papel da tecnologia da informação (ti) na estratégia das organizações. *Gestão & Produção*, 8(2):160–179, 2001.
- [47] James D Thompson and Frederick L Bates. Technology, organization, and administration. *Administrative Science Quarterly*, pages 325–343, 1957.
- [48] Carol V Brown. Examining the emergence of hybrid is governance solutions: Evidence from a single case site. *Information systems research*, 8(1):69–94, 1997.
- [49] Peter Weill and Marianne Broadbent. *Leveraging the new infrastructure: how market leaders capitalize on information technology*. Harvard Business Press, 1998.
- [50] Pieter MA Ribbers, Ryan R Peterson, and Marilyn M Parker. Designing information technology governance processes: diagnosing contemporary practices and competing theories. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pages 3143–3154. IEEE, 2002.

- [51] Andrew C Boynton, Gerry C Jacobs, and Robert W Zmud. Whose responsibility is it management? *MIT Sloan Management Review*, 33(4):32, 1992.
- [52] Petter Gottschalk. Managing it functions. *Strategies for information technology governance*, page 246, 2004.
- [53] Mathias Sallé. It service management and it governance: review, comparative analysis and their impact on utility computing. *Hewlett-Packard Company*, pages 8–17, 2004.
- [54] Peter Weill, Mani Subramani, and Marianne Broadbent. It infrastructure for strategic agility. 2002.
- [55] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, 40(2):5, 2008.
- [56] C LAUDON KENNETH and JANE P LAUDON. Sistemas de informação gerenciais. *Editora Person. São Paulo*, 2011.
- [57] Marcos Pitanga. *Construindo supercomputadores com Linux*. Brasport, 2004.
- [58] John L Hennessy and David Patterson. *Arquitetura de computadores: uma abordagem quantitativa*, volume 5. Elsevier Brasil, 2014.
- [59] BN JOÃO. Sistemas de informação, 2012.
- [60] Anthony I Wasserman. How the internet transformed the software industry. *Journal of Internet Services and Applications*, 2(1):11–22, 2011.
- [61] Chris DiBona and Sam Ockman. *Open sources: Voices from the open source revolution*. "O'Reilly Media, Inc.", 1999.
- [62] Lothar Determann. Dangerous liaisons-software combinations as derivative works-distribution, installation, and execution of linked programs under copyright law, commercial licenses, and the gpl. *Berkeley Tech. LJ*, 21:1421, 2006.
- [63] Andrew M St Laurent. *Understanding open source and free software licensing: guide to navigating licensing issues in existing & new software*. "O'Reilly Media, Inc.", 2004.
- [64] Lawrence Rosen. *Open source licensing: Software freedom and intellectual property law*. Prentice Hall PTR, 2004.
- [65] Robert Gobeille. The fossology project. In *Proceedings of the 2008 international working conference on Mining software repositories*, pages 47–50. ACM, 2008.
- [66] Motoei Azuma. Software products evaluation system: quality models, metrics and processes—international standards and japanese practice. *Information and Software Technology*, 38(3):145–154, 1996.
- [67] Normas Técnicas. Abnt-associação brasileira de normas técnicas. 1999.
- [68] IGG CAPOVILLA. *Elementos Intrínsecos do Software e sua Influência na Qualidade do Processo de Desenvolvimento*. 1999. 108 P. PhD thesis, Dissertação (Mestrado em Qualidade)–IMECC-Instituto de Matemática, Estatística e Computação Científica, UNICAMP-Universidade Estadual de Campinas, Campinas-SP, 1989.



- [69] Regina Colombo and Ana Guerra. The evaluation method for software product. In *Proc of the 15th Int Conf on Softw & Syst Eng & Appl, Paris, France, December*, pages 3–4, 2002.
- [70] International Organization for Standardization and International Electrotechnical Commission. *Software Engineering–Product Quality: Quality model*, volume 1. ISO/IEC, 2001.
- [71] James A Redmond and Reynold Ah-Chuen. Software metrics—a user’s perspective. *Journal of Systems and Software*, 13(2):97–110, 1990.
- [72] Shari Lawrence Pfleeger and JC Fitzgerald Jr. Software metrics tool kit: support for selection, collection and analysis. *Information and software technology*, 33(7):477–482, 1991.
- [73] Haijun Yang. Measuring software product quality with iso standards base on fuzzy logic technique. In *Affective Computing and Intelligent Interaction*, pages 59–67. Springer, 2012.
- [74] Pei-Yu Chen and Lorin M Hitt. Measuring switching costs and the determinants of customer retention in internet-enabled businesses: A study of the online brokerage industry. *Information systems research*, 13(3):255–274, 2002.
- [75] Paul Klemperer. Markets with consumer switching costs. *The quarterly journal of economics*, 102(2):375–394, 1987.
- [76] Joseph Farrell and Paul Klemperer. Coordination and lock-in: Competition with switching costs and network effects. *Handbook of industrial organization*, 3:1967–2072, 2007.
- [77] Paul Klemperer. Competition when consumers have switching costs: An overview with applications to industrial organization, macroeconomics, and international trade. *The review of economic studies*, 62(4):515–539, 1995.
- [78] Wim Van Grembergen. *Strategies for information technology governance*. IGI Publishing, 2003.
- [79] D.E. Comer. *Redes de Computadores E Internet 4 Ed*. Bookman, 2007.
- [80] Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martín Casado, Nick McKeown, and Scott Shenker. Nox: towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*, 38(3):105–110, 2008.
- [81] James Kempf, Scott Whyte, Jonathan Ellithorpe, Peyman Kazemian, Mart Haitjema, Neda Beheshti, Stephen Stuart, and Howard Green. Openflow mpls and the open source label switched router. In *Proceedings of the 23rd International Teletraffic Congress*, pages 8–14. International Teletraffic Congress, 2011.
- [82] Bruce S Davie and Adrian Farrel. *MPLS: next steps*, volume 1. Morgan Kaufmann, 2008.
- [83] Celio Trois, Daniel Weingaertner, Diego Pasqualin, Edemir Maciel, Eduardo Almeida, Fabiano Silva, Hegler Tissot, Luis CE Bona, Marcos Castilho, Marcos Didonet, et al. Transparency meets management: a monitoring and evaluating tool for governmental projects. In *Computer Systems and Applications (AICCSA), 2017 IEEE/ACS 14th International Conference on*, pages 1429–1435. IEEE, 2017.

- [84] GNU General Public License. Free software foundation. *License used by the Free Software Foundation for the GNU Project. See <http://www.fsf.org/copyleft/gpl.html> (<http://www.fsf.org/copyleft/gpl.html>), 1991.*
- [85] Ralph Kimball and Margy Ross. *The data warehouse toolkit: the complete guide to dimensional modeling*. John Wiley & Sons, 2011.